الكتاب الاكثر مبيعا فى العالم



COMPREHENSIVE
AUTHORITATIVE
WHAT YOU NEED

Master the simple yet powerful new markup language that's revolutionizing the web
Build well-formed sensibly organized web documents
Create entirely new markup languages to fit your own needs

إكس إم إل

XML™ Bible



تأليف
إليوت رستى هارولد

الترجمة بإعتماد
د / خالد العامرى

إكس إم إل

بايبل

XML

Bible

دار الفاروق للنشر والتوزيع

أكبر مركز فى الشرق الأوسط

لإصدار أحدث الكتب فى عالم الكمبيوتر

العنوان

فرع وسط البلد: ٣ شارع منصور - المبتديان - متفرع من شارع مجلس الشعب محطة مترو سعد زغلول - القاهرة - مصر.

تليفون : ٧٩٥٣٠٣٢ (٠٠٢٠٢) - ٧٩٤٣٢٠٣ (٠٠٢٠٢)

فاكس : ٧٩٤٣٦٤٣ (٠٠٢٠٢)

فرع الدقى: ١٢ شارع الدقى الدور السابع - إتجاه الجامعة منْزَل كوبرى الدقى

تليفون : ٣٣٨١٠٢٢ (٠٠٢٠٢)

فاكس : ٣٣٨٢٠٧٤ (٠٠٢٠٢)

الطبعة العربية الأولى ٢٠٠٠

عدد الصفحات: ١٢٠٨ صفحة

رقم الإيداع: ٣١٢٧ لسنة ٢٠٠٠

الترقيم الدولى: 977-307-094-8

إكس إم إل
بايبل

XML
Bible

تأليف

اليوت راستي هارولد

إعداد وترجمة

دار الفاروق وجي إن إس

لمحة على المحتويات

المحتويات

٢ مقدمة إلى تطبيقات XML ٥٥

٣ إنشاء أول مستند باستخدام XML ٩١

٤ هيكلة البيانات ١٠٣

٥ السمات والعلامات الفارغة وXSL ١٤٧

الجزء الثاني: تعريفات نوع المستند

الجزء الثالث: لغات الأنماط

١٣ أوراق نمط المستوى ٢ ٥٠٧

١٥ كائنات تنسيق XML ٦٥٥

الجزء الرابع: تقنيات إضافية

١٧ مؤشرات XPointers ٧٤٧

١٨ Namespace ٧٧٧

١٩ هيكل وصف المورد ٧٩٥

الجزء الخامس: تطبيقات XML

٢٠ قراءة تعريفات نوع المستند ٨٢٥

٢٢ لغة الترميز المحورية ١٠٠٩

٢٣ تصميم تطبيق XML جديد ١٠٤٣

المقدمة

مما لا شك فيه أن لغة XML تعد واحدة من أهم التقنيات البرمجية لكتابة الصفحات والتي أثرت شبكة الإنترنت منذ ظهور لغة Java وحتى هذه اللحظة بما أضافته على تصميم المواقع من سهولة ويسر وقوة ومتعة.

تم إعداد هذا الكتاب ليكون مقدمة إلى عالم XML السريع التطور والفعال ولهذا يوضح الكتاب كيفية كتابة مستندات XML واستخدام أوراق النمط لتحويل هذه المستندات إلى لغة HTML لتستطيع المستعرضات الموجودة قراءتهم. كما يوضح كيفية استخدام تعريفات نوع المستند (DTDs) لوصف المستندات والتأكد من صحتهم وتتزايد أهمية ذلك كلما قدمت مستعرضات مثل Mozilla و Internet Explorer 5.0 دعم أصلي للغة XML.

القارئ

يختلف كتاب XML Bible عن بقية كتب XML الموجودة بالأسواق لأن كتاب XML Bible يغطي XML ليس من منظور المبرمجين ولكن من منظور ناشر صفحات الويب ولهذا فإن الكتاب لا يناقش صيغ BNF النحوية أو شجرة توزيع العناصر باستفاضة ولكن يعلمنا كيفية استخدام XML والأدوات المتاحة الآن لإنتاج مواقع ويب يسهل استخدامها وتعديلها وتكون مثيرة وجذابة لتجعل مستخدم الموقع يعود إليها مرة أخرى.

هدف هذا الكتاب هم مطوري مواقع الويب حيث نفترض في الكتاب إن القارئ يريد استخدام XML لإنشاء مواقع ويب يكون من الصعب أو المستحيل إنشائها بواسطة HTML وسيندهش المستخدم عند اكتشاف كيف تستطيع لغة XML بالتعاون مع أوراق النمط وبعض الأدوات الأخرى إنجاز أشياء كانت تتطلب فيما سبق إما برامج متخصصة تكلف مئات بل الآلاف من الدولارات يتحملها كل مبرمج أو تتطلب معلومات موسعة عن لغات البرمجة مثل Perl أما بالنسبة للبرامج في هذا الكتاب فلن تكلف أكثر من دقائق لتحميلها ولا تتطلب أي من الخطوات المراد تنفيذها أي قدر من البرمجة.

ما يجب معرفته

تبني لغة XML على لغة HTML والبنية المضمنة في الإنترنت ولهذا العرض نفترض في هذا الكتاب معرفة القارئ كيفية استخدام FTP وإرسال البريد الإلكتروني وتحميل URLs في

مستعرض الويب وكذلك معرفته بلغة HTML وصولاً إلى المستوى الـــذي يدعمـــه Netscape 1.1، ومن جهة أخرى عندما يقوم الكتاب بتغطية أوجه حديثة بالنسبة للغـــة HTML و التـــي لـــم ينتشر استخدامها بعد مثل أوراق النمط المتتالي تتم تغطية تلك الموضوعات بتوسع و استفاضة.

لتكون أكثر تحديداً يفترض هذا الكتاب معرفة القارئ بما يلي:

- كتابة صفحة HTML أساسية متضمنة الارتباطات والصور والنص باســـتخدام محـــرر نصوص.
- وضع تلك الصفحة على خادم ويب.

ومن جهة أخرى لا يفترض هذا الكتاب ما يلي:

- معرفة القارئ بلغة SGML وفي الواقع فإن مقدمة هذا الكتاب هي الجزء الوحيـــد فـــي الكتاب الذي تستخدم فيه كلمة SGML حيث تعتبر لغة XML أبسط وأكثر انتشاراً من لغة SGML.
- إن القارئ مبرمج سواء باستخدام لغة Java أو Perl أو C أو أي لغات أخـــرى حيـــث تعتبر XML لغة ترميز وليس لغة برمجة ولذلك فالقارئ ليس فـــي حاجـــة لأن يكـــون مبرمجاً لكتابة مستندات بلغة XML.

نتعلم من الكتاب

هدف الكتاب الرئيسي هو تعليم القارئ كيفية كتابة مستندات بلغة XML للويب ومن حسن الحـــظ فإن للغة XML منحنى تعلم مستو مثل لغة HTML على عكس لغة SGML. وهذا يعني أنه كلمـــا تعلمت شيئاً استطعت إنجازه بنجاح ولهذا تعتمد فصول هذا الكتاب على بعضها البعض ولابد مـــن قراءتها بالترتيب حيث تتعلم أثناء القراءة ما يلي:

- كيف يتم إنشاء مستند XML وتوصيله إلى القارئ.
- كيف تجعل عملية كتابة مستندات XML أسهل في التطوير والحفاظ عليها عن مثيلتها لغة HTML.
- كيف ترسل مستندات XML على خادم ويب في نموذج يستطيع الكل قراءته.
- كيفية التأكد من جودة تكوين مستند XML.
- كيفية استخدام الحروف الدولية
- كيفية التأكد من صلاحية مستندات\بها DTDs.
- كيفية استخدام كينونات لبناء مستندات ضخمة من أجزاء أصغر.



- كيف تصف السمات البيانات.
- كيفية العمل مع بيانات غير بيانات XML.
- كيفية تنسيق المستندات باستخدام أورق نمط CSS وXML.
- كيفية ربط المستندات باستخدام ارتباطات xLinks ومؤشرات Xpointers.
- كيفية دمج مفردات XML المختلفة مع حيز الأسماء namespaces.
- كيفية كتابة البيانات المفصلة لصفحات ويب باستخدام RDF.

يحتوي الجزء الأخير من هذا الكتاب على عدة أمثلة عملية للغةXML تستخدم فيها التطبيقات الفعلية بما في ذلك:

- تصميم موقع ويب
- تقنية دفع المعلومات Pushالرسوم الاتجاهية Vector Graphics
- سلسلة النسب (الأصل) Genealogy

طريقة تنظيم الكتاب

ينقسم هذا الكتاب إلى خمسة أجزاء ويتضمن ثلاثة ملاحق:

١ – تقديم XML.

٢ – تعريفات نوع المستند.

٣ – لغات الأنماط.

٤ – تقنيات إضافية.

٥ – تطبيقات XML.

يمكن استخدام لغة XML لإنشاء صفحات ويب مؤثرة بعد الانتهاء من قراءة الكتاب وما يلي هو شرح للأجزاء الخمسة والثلاث ملاحق:

الجزء الأول: تقديم XML

يحتوي الجزء الأول على الفصول من ١ إلى ٧. يبدأ هذا الجزء في توضيح تاريخ ونشأة لغةXML والأهداف المراد تحقيقها من استخدامها كما يوضح كيف تتلاءم الأجزاء المختلفة من معادلة XML مع بعضها البعض لإنشاء وتوصيل المستندات إلى القارئ. يقدم الجزء الأول أيضاً عدة أمثلة لتطبيقات XML لتوضيح مدى سعة تطبيق XML بما في ذلك Vector Markup

Language (VML), Resource Description Framework (RDF), Mathematical Markup Language (MathML), the Extensible Forms Description Language (XFDL). حيث نتعلم عن طريق تلك الأمثلة كيفية كتابة مستندات XML بعلامات يقوم المستخدم بتعريفها للمستند وكيفية تحرير تلك العلامات باستخدام محرر النص و إلحاق أوراق بيانات إليهم وتحميلهم إلى مستعرض ويب مثل Internet Explorer 5.0 أو Mozilla وفي النهاية نتعلم كيف يتم كتابة مستندات XML بلغات غير اللغة الإنجليزية مثل الصينية و العربية و الروسية.

الجزء الثاني: تعريفات نوع المستند

يحتوي الجزء الثاني على الفصول من ٨ إلى ١١. تركز تلك الفصول على تعريفات نوع المستند (DTDs). قد يحتوي مستند XML بصورة اختيارية على DTD تقوم بتحديد العناصر التي يسمح بوجودها والعناصر التي لا يسمح بوجودها في مستند XML كما تحدد DTD البنية و السياق الصحيح لهذه العناصر ليمكن لموزع الصلاحية قراءة مستند ومقارنته إلى DTD الخاص به ليظهر أي خطأ يكتشفه ويتيح DTDs لناشري المستند التأكد أن عملهم يلبي احتياجات أي معايير ضرورية.

نتعلم في الجزء الثاني كيفية إلحاق DTD بمستند والتحقق من صحة المستندات بمقارنتها بتعريفات نوع المستند DTDs الخاصة بهم وكتابة DTDS الخاصة بالقارئ. كما نتعلم أيضاً بناء جملة إعلان العناصر والسمات والكينونات ومجموعة الرموز ويوضح الكتاب كيفية استخدام تعريفات الكينونات ومراجع الكينونة لبناء كل من المستند وDTD من أجزاء متعددة مستقلة وينتج عن هذا جعل المستندات الطويلة التي يصعب تتبعها أبسط عن طريق فصلهم إلى مكونات ووحدات نمطية متصلة. في نهاية هذا الجزء نتعلم كيف يتم دمج أنواع أخرى من البيانات مثل ملفات صور GIF ونص أولي في مستند XML.

الجزء الثالث: لغات النمط

يتضمن الجزء الثالث الفصول من ١٢ إلى ١٥.يوضح هذا الجزء كيف أن ترميز XML فقط هو الذي يحدد ما الذي يوجد في مستند وعلى عكس لغة HTML فأنه لا يوضح كيف يبدو المحتوى. يتم تخزين المعلومات الخاصة بمظهر مستند XML عند طباعته وعرضه في مستعرض ويب في ورقة نمط ويمكن استخدام أوراق نمط مختلفة لنفس المستند فعلى سبيل المثال يمكن استخدام ورقة نمط تحدد الخط الصغير للطباعة وورقة نمط أخرى تستخدم نوع خط أكبر للاستخدام على الشاشة وثالثة تحتوي على خط humongous مطلق لوضع المستند على الحائط في ندوة وتستطيع باستخدام أوراق نمط مختلفة تغيير مظهر مستند XML وذلك دون الاقتراب من المستند نفسه.

يوضح الجزء الثالث بالتفصيل استخدامات لغات ورقتي النمط Cascading Style Sheets(CSS) وExtensible Style Language (XSL) على الويب.

تعتبر (CSS) لغة ورقة نمط بسيطة مصممة للاستخدام مع لغة HTML وتوجد CSS في إصدارين هما CSS Level 2 وCSS Level 1 تقدم ورقة نمط CSS Level 1 معلومات أساسية بشأن الخطوط والألوان وتعيين موضع وخصائص النص كما يتم دعمها بواسطة مستعرضات الويب الحالية HTML وXML أما بالنسبة لورقة نمط CSS Level 2 فهي قياس حديث يضيف دعم لأوراق النمط السمعية وأنماط واجهة المستخدم والنص الدولي ونص اللغة العربية وتعتبر CSS قياس بسيط نسبياً يطبق قواعد نمط ثابتة لمحتويات عناصر محددة.

ومن جهة أخرى تعتبر لغة XSL لغة نمط أكثر تعقيدا وأكثر فاعلية حيث أنها لا تطبق فقط أنماط على محتويات العناصر ولكن أيضا تعيد ترتيب العناصر وتضيف نص متداول وتحول المستندات بطريقة عشوائية. تنقسم لغة XSL إلى جزأين هما لغة التحويل وذلك لتحويل شجرة XML إلى شجرة بديلة ولغة تنسيق لتحديد مظهر العناصر لشجرة XML وفي الوقت الحالي تدعم لغة التحويل معظم الأدوات المستخدمة بصورة أفضل من تدعيمها للغة التنسيق وعموماً فقد بدأ Internet Explorer 5.0 الخاص بمايكروسوفت في تدعم لغة التحويل بالإضافة إلى محركات تنسيق ثالثة.

الجزء الرابع: تقنيات إضافية

يحتوي الجزء الرابع على الفصول من ١٦ إلى ١٩. يقدم هذا الجزء بعض لغات وبناء جمل XML الأساسية والتي تعتمد على أساسيات لغة XML. تقدم ارتباطات X (XLINKS) ارتباطات نص مترابط متعددة الاتجاهات وهي أكثر قوة من علامة HTML البسيطة التي تخص <A>. تقدم مؤشرات Xpointers بناء جملة جديدة يمكن إلحاقه بنهاية URLs ليربط ليس فقط مستندات محددة ولكن ليربط أجزاء معينة من مستندات محددة وتستخدم مسافات الأسماء البادئات prefixes وURLs لتفادي الغموض والمشاكل التي تنتج من تعارضات لغات ترميز XML. يعتبر Resource Description Framework (RDF) تطبيق في XML يستخدم لتضمين بيانات مفصلة في مستندات XML وHTML والبيانات المفصلة هي معلومات بشأن المستند مثل الناشر والتاريخ وعنوان العمل بدلا من تفاصيل العمل نفسه ويمكن إضافة كل ذلك للغات الترميز التي تعتمد على XML لزيادة قدرتهم وفائدتهم.

الجزء الخامس: تطبيقات XML

يوضح الجزء الخامس الذي يضم الفصول من ٢٠ إلى ٢٣ أربعة استخدامات عملية للغة XML في مجالات مختلفة. أولا XHTML وهو إعادة صياغة للغة HTML 4.0 والتي ينتج عنها مستندات XML صالحة. ثانياً Channel Definition Format (CDF) الخاصة بمايكروسوفت وهي لغة ترميز تعتمد على XML لتعريف القنوات التي تدفع محتويات مواقع الويب التي تم تحديثها للمشتركين. ثالثا Vector Markup Language (VML) وهو تطبيق XML يستخدم للرسومات التي يتم تغيير حجمها والتي تستخدمها كل من برامج مايكروسوفت أوفيس ٢٠٠٠ وInternet Explorer 5.0 وفي النهاية يتم تطوير تطبيق جديد كلية للبيانات المنسوبة لأصلها لتوضيح ليس فقط كيفية استخدام علامات XML ولكن لماذا ومتى يتم اختيارهم.

الملاحق

يركز الملحقان الخاصان بالكتاب على المواصفات الرسمية الخاصة بلغة ترميز XML على عكس باقي الكتاب الذي استخدم فيه وصف غير رسمي. يقدم ملحق أ توضيح مفصل لثلاثة أجزاء فردية لمواصفات XML 1.0 وهم تدقيق XML النحوي في BNF وشروط جودة التكوين وشروط الصلاحية. يحتوي الملحق ب على مواصفات W3C XML 1.0 الرسمية التي نشرت بواسطة W3C. ويشرح الملحق ج محتويات الأقراص المضغوطة المصاحبة للكتاب.

ما تحتاجه

تحتاج للاستفادة القصوى من هذا الكتاب وXML إلى ما يلي:

- كمبيوتر شخصي نظام تشغيله يكون ويندوز ٩٥ أو ويندوز ٩٨ أو ويندوز NT
- Internet Explorer 5.0
- برنامج Java 1.1 أو برنامج تشغيل أحدث

أي نظام يعمل باستخدام الويندوز يستطيع تشغيل XML ونفترض في هذا الكتاب استخدام إما ويندوز ٩٥ أو NT 4.0 أو أي نظام أحدث على غرار لغة Java فإن لغة XML تعتبر نظام تشغيل مستقل وعلى الرغم من أن تعليمات XML البرمجية هي نص خالص يمكن كتابته بواسطة أي محرر فإن العديد من الأدوات متاحة حالياً فقط على برامج الويندوز.

مع ذلك وعلى الرغم من أنه لم يعد هناك العديد من برامج XML الأصلية تعمل مع أنظمة التشغيل Macintosh Unix فأن هناك عدد متزايد من برامج XML التي تكتب باستخدام لغة Java فإذا كان نظام تشغيل الجهاز له Java 1.1 أو أي برنامج تشغيل أحدث فأن ذلك سيفي بالغرض حتى وإن لم تستطع تحميل مستند XML مباشرة إلى مستعرض ويب فمازلت تستطيع تحويل المستندات إلى مستندات XML وعرضهم. وعند إصدار Mozilla فإنها يجب أن تقدم أفضل مستعرض XML على الإطلاق عبر أنظمة التشغيل المتعددة.

كيفية استخدام هذا الكتاب

تم تصمم هذا الكتاب لتتم قراءته بأكمله فكل فصل يبني على ما سبق وبالطبع يمكن التغاضي عن الأجزاء التي تعرفها ومن الأفضل أثناء قراءة الكتاب القيام بتجربة بعض الأمثلة وكتابة بعض مستندات XML وقبل البدء يجب مراجعة الملاحظات التالية عن المصطلحات النحوية المستخدمة في هذا الكتاب.

علي عكس HTML فاستجابة XML تختلف بالنسبة للأحرف اللاتينية الكبيرة والصغيرة فمثلا <FATHER> تختلف عن <father> أو <Father> وذلك لأن عنصر father يختلف عن عنصر FATHER أوfather وللأسف فإن لغات ترميز حالات الأحرف "سواء أحرف لاتينية كبيرة أو صغيرة" تتعارض مع الاستخدام المعتاد للغة الإنجليزية وهذا يعني أنه قد تجد كلمات لا تبدأ بحرف استهلالي كبير وقد ترى حروف استهلالية كبيرة في وسط الجملة وعموماً وضعت معظم التعليمات البرمجية الخاصة بلغات HTML وXML المستخدمة في هذا الكتاب بخط متباعد تباعد أحادي ولذا ففي معظم الأحيان يتضح من السياق ما هو المعنى المراد.

تم وضع علامات الترقيم داخل علاقات الاقتباس عندما تتصل بالمادة المنقولة وهذا هو الأسلوب البريطاني في الكتابة حيث يعتبر أكثر منطقية خاصة عند التعامل مع تعليمات المصدر البرمجية حيث يحدث الاختلاف ما بين وجود الفاصلة والنقطة أو عدم وجود علامات ترقيم نهائياً اختلافاً ما بين تعليمات برمجية صحيحة وأخرى خاطئة.

١

الفصل ٧

اللغات الأجنبية والنص غير الروماني

مقدمة عن XML



الفصل

* * * * *

في هذا الفصل

مقدمة عن XML

يقوم هذا الفصل بتقديم XML ويشرح ما هو XML بشكل عام وكيفية استخدامه ويعـرض كيـف تلائم الأجزاء المختلفة لمعادلة XML بعضها البعض وكيفية إنشاء مستندات XML وعرضها علـى القراء.

ما هو XML

ترمـز XML إلـى Extensible Markup Language "فـي أغلـب الأحيـان تكتـب eXtensibleMarkup Language حتى تتوافق مع التسمية المختصـرة" وتتكـون XML مـن مجموعة قوانين معدة لتعريف علامات خاصة بالمعاني والرموز وتقسيمها إلى أجـزاء وتحديـد الأجزاء المختلفة في المستند وهي لغة ترميز البيانات التفصيلية التـي تحـدد صيغـة الجملـة المستخدمة في تعريف اللغات الهيكلية الرمزية الخاصة بمجالات محددة أخرى.

XML لغة ترميز بيانات تفصيلية

أول شئ يجب توضيحه عن XML هو أنها ليست لغة ترميز أخرى مثـل لغـة ترمـيز النـص المترابط Hypertext Markup Language (HTML) وتقوم هذه اللغات بتعريـف مجموعـة محدودة من العلامات التي تصف عدد محدود من العناصر، وإذا كانت اللغة التي تسـتخدمها لا تحتوي علي العلامات التي تحتاجها، إذا فقد جانبك الحظ. يمكنك انتظار الإصدار التالي من لغـة الترميز أملاً في أن تحتوي علي العلامات التي تحتاجها ولكنك عندها ستكون تحت أمر ما يختار العميل تضمينه.

ومع ذلك فأن XML هي لغة ترميز بيانات تفصيلية أي لغة تقوم فيها بإعداد العلامـات التـي تحتاجها ويجب تنظيم هذه العلاقات طبقاً لقواعد محددة لكنها مرنة فمثلا إذا كنت تعمل في علـم الأنساب وتحتاج لوصف الناس وميلادهم ووفياتهم وأماكن الدفن وعائلاتهم وزيجاتهم وطلاقـهم وهكذا يمكنك إنشاء علامات لكل هؤلاء. لن تحتاج إلى جعل بياناتك تلائـم الفقـرات أو تكتـب عناصر في قائمة أو أي فئة عامة أخرى.

العلامات التي أنشأتها يمكنك توثيقها في Document Type Definition (DTD). يمكنـك معرفة المزيد عن DTD في الجزء الثاني من هذا الكتاب ولكن سنستخدم DTD حاليـاً كمفـرد لغوي وكبناء جملة لأنواع محددة من المستندات فمثـلاً يقـوم MOL.DTD فـي لغـة Peter Murray-Rust وهي لغة ترميز كيميائية Chemical Markup Language التي يرمـز لـها (CML) بوصف المفردات اللغوية وبناء الجملة للعلوم الجزئية مثل الكيميـاء وعلـم البلـورات وتتضمن علامات للذرات والجزئيات...الخ. يمكن أن يشترك مجموعة من العـاملين فـي حقـل العلوم الجزئية في DTD معينة وتوجد بعض تعريفات DTD متاحة في المجالات الأخرى كمـا يمكنك أيضاً إنشاء DTD الخاص بك.

تقوم XML بتعريف بناء جملة بيانات تفصيلية يجب أن تتبعها لغات ترميز مثـــل MusicML وMathML و(CML). إذا كان التطبيق يفهم بناء الجملة هذا فإنه يفهم كـــل اللغـــات التـــي تـــم صياغتها من لغة البيانات التفصيلية هذه ولا يحتاج المستعرض إلى معرفة كل علامة قد تســتخدم بواسطة لغات الترميز المختلفة ولكن كبديل يقوم باستكشاف العلامات المســـتخدمة بواســـطة أي مستند حيث أنه يقرأ المستند نفسه أو DTD الخاصة به. التعليمات الكاملة عـــن كيفيـــة عـــرض محتويات هذه العلاقات يتم وضعها في ورقة نمط منفصلة مرتبطة بالمستند

فمثلا معادلة Schrodinger :

$$i\hbar\frac{\partial\psi(r,t)}{\partial t}=-\frac{\hbar^2}{2m}\frac{\partial^2\psi(r,t)}{\partial x^2}+V(r)\,\psi(r,t)$$

نجد أن الأوراق العلمية مليئة بمثل هذه المعادلات ولكن انتظر العلماء ثمان سنوات حتـــى يقـــوم المستعرض بتدعيم العلامات المستخدمة في كتابة المعادلات الرياضية الأساسية. أيضـــاً انتظـــر الموسيقيون لأن Netscape Navigator وInternet Explorer لم يدعموا العلامات الموسيقية.

والغرض من وجود XML هو أنك لن تكون بحاجة إلى انتظار أن يدعم المستعرض ما تريد حيث يمكنك اختراع العلامات التي تحتاجها وأن تخبر المستعرض كيف يعرض هذه العلامات.

XML لغة تصف البناء والمعاني وليس التنسيق

ثاني شئ يجب توضيحه عن XML هو أنها لغة ترميز تصف بنية المستند ومعناه ولكنها لا تصف تنسيق العناصر في الصفحة. يمكن إضافة التنسيق إلى مستند مع ورقة النمط ويحتـــوي المســـتند نفسه على علامات تحديد محتوى المستند وليس كيف يبدو المستند.

على النقيض فإن HTML يشمل التنسيق والبناء وتركيب الجملة. وتعد <B> علامة التنســـيق التي تجعل المحتوى يكتب بالخط الأسود العريض بينما <STRONG> هـــي علامـــة خاصـــة بالمعنى وهذا يعني إن المحتوى على درجة كبيرة من الأهمية و<TD> هو علامة البنيـــة التـــي تشير أن المحتوى هو خلية في جدول وفي بعض الأحيان قد يكون لدى بعض العلامات الأنـــواع الثلاثة من المعنى فمثلا علامة <H1> يمكن أن تعني خط Helvetica أسود عريض بحجـــم ٢٠ نقطة وكذلك تعني رأس الصفحة من المستوى الأول وعنوان الصفحة.

فمثلا في HTML يمكن وصف أغنية باستخدام عنوان تعريف وبيانات تعريف وقائمـــة غـــير مرتبة وعناصر قائمة ولكن كل هذه العناصر لا علاقة لها بالموسيقى وســـيبدو HTML بالشـــكل التالي:

```
<dt>Hot Cop
<dd> by Jacques Morali, Henri Belolo, and Victor Willis
<ul>
<li>Producer: Jacques Morali
<li>Publisher: PolyGram Records
<li>Length: 6:20
<li>Written: 1978
<li>Artist: Village People
</ul>
```

في XML سيبدو ترميز نفس البيانات هكذا:

```
<SONG>
  <TITLE>Hot Cop</TITLE>
  <COMPOSER>Jacques Morali</COMPOSER>
  <COMPOSER>Henri Belolo</COMPOSER>
  <COMPOSER>Victor Willis</COMPOSER>
  <PRODUCER>Jacques Morali</PRODUCER>
  <PUBLISHER>PolyGram Records</PUBLISHER>
  <LENGTH>6:20</LENGTH>
  <YEAR>1978</YEAR>
  <ARTIST>Village People</ARTIST>
</SONG>
```

بدلا من العلامات العامة مثل <dt> و في XML يتم استخدام علامات مفهومة مثل <SONG> <TITLE> و<COMPOSER> وفي هذا العديد من المزايا تتضمن سهولة قراءة التعليمات البرمجية بالنسبة للقارئ ليستطيع تحديد ما الذي يقصده الناشر.

كذلك تقوم لغة ترميز XML بتسهيل مهمة برامج إحضار صفحات الإنترنت في تحديد موقع كل من الأغاني في مستندك. أما في HTML فإن البرنامج لا يستطيع إخبارك بأكثر من أن العنصر هو dt ولا يمكنهم تحديد ما إذا كان dt يمثل عنوان أغنية أو تعريف أو هو مجرد أحد وسائل المصمم المفضلة لترك مسافة بادئة في النص ومن الممكن أن تجد مستند واحد يحتوي على عناصر dt بمعانيه الثلاث.

يمكنك اختيار أسماء عنصر XML حتى يكون لها معاني إضافية في سياقات أخرى فمثلا ربما كانت أسماء الحقول لقاعدة بيانات. يعتبر XML أكثر مرونة وأكثر استخداماً من HTML لأن العدد المحدود من العلامات لا يمكن استخدامه لتحقيق أغراض مختلفة.

سر سعادة المبرمجين بوجود XML

لأن XML يجعل مهام تطوير الويب أسهل والتي كانت من قبل أصعب بكثير عند استخدام HTML فقط حيث يجعل "XML" المهام التي كانت تبدو مستحيلة ممكن إنجازها ولأن XML قابل للتطوير وزيادة استخداماته فإن المطورين يحبونه للعديد من الأسباب وتختلف الأسباب تبعاً للاحتياجات الشخصية إلا أنه بمجرد إلمامك الكامل بهذه اللغة ستجد أن هناك أكثر من حل للمشكلة التي كانت تواجهك وهذا المقطع يشرح بعض من الاستخدامات العامة للغة XML التي تسعد المبرمجين وفي الفصل الثاني ترى بعض التطبيقات المحددة التي تم تطويرها باستخدام XML.

تصميم لغات الترميز المحددة المجال

يسمح XML لمهن متعددة "مثل الموسيقى والكيمياء والرياضة" بتطوير لغات الترميز المحددة المجال وهذا يسمح للعاملين في المجال بتبادل الملحوظات والبيانات والمعلومات دون القلق من أن الأشخاص الذين قاموا باستلام البيانات يملكون نفس البرامج التي استخدمت في إنشاء البيانات بل ويمكنهم إرسال المستندات إلى أشخاص خارج المهنة مع التأكد من أن لديهم القدرة على عرض المستندات.

بالإضافة إلى ذلك لا يؤدي إنشاء لغات ترميز لمجالات معينة إلى تعقيد الأمور لهؤلاء الأشخاص الذين يعملون خارج المهنة وربما لا تكون مهتماً بالرسوم الهندسية الكهربائية ولكن بالنسبة للمهندسين الكهربائيين يكون ذلك مهماً. وربما لا تحتاج إلى إضافة ورقة موسيقى إلى صفحات الويب الخاصة بك ولكن بالنسبة للملحنين فإن ذلك مهما جدا ويسمح XML لمهندسي الكهرباء بوصف دوائرهم ويسمح للملحنين بالكتابة دون التخطي على غيرهم، ولن يحتاج أي من الحقلين إلى دعم خاص من المستعرضات أو إلى برامج مضافة معقدة كما هو الحال اليوم.

بيانات الوصف الذاتي

أغلب بيانات الكمبيوتر خلال الأربعين عام الأخيرة قد فقدت وذلك بسبب إما كوارث طبيعية أو فساد وسائط النسخ الاحتياطي "وتلك مشكلة أيضاً يصعب على XML حلها" حيث لم يهتم أحد بتوصيف كيفية قراءة بيانات الوسائط وتنسيق المستندات فمثلا ملف لوتس 3-2-1 على قرص مرن حجمه ٥,٢٥ بوصة يكون من الصعب استرجاع محتوياته في أغلب الشركات بدون استثمار الكثير من الوقت والموارد. أما البيانات التي توجد في تنسيق ثنائي مثل Lotus Jazz، قد تكون فقدت إلى الأبد.

تعتبر لغة XML على المستوى الأساسي تنسيق بيانات بسيط ويمكن كتابتها في نص ASCII خالص ١٠٠% بالإضافة إلى بعض التنسيقات المعرفة بصورة جيدة ويتميز نص ASCII بمقاومته لاحتمالات التلف بصورة معقولة وإزالة أي عدد من البايتات لا يؤثر على بقية النص وهذه الخاصية غير موجودة في الكثير من الصيغ الأخرى مثل البيانات المضغوطة أو كائنات Java المسلسلة حيث فساد أو ضياع بايت واحد يمكن أن يجعل بقية الملف غير مقروء.

وعلى مستوى أعلى يقوم XML بشرح نفسه. على سبيل المثال، نفترض أنك عالم آثار في القرن ٢٣ وواجهك هذا الكم من تعليماتXML البرمجية على قرص مرن قديم من مخلفات الزمن الماضي:

```
<PERSON ID="p1100" SEX="M">
  <NAME>
    <GIVEN>Judson</GIVEN>
    <SURNAME> McDaniel</SURNAME>
  </NAME>
  <BIRTH>
    <DATE>21 Feb 1834</DATE>  </BIRTH>
  <DEATH>
    <DATE>9 Dec 1905</DATE>  </DEATH>
</PERSON>
```

حتى إذا لم يكنXML مألوفا لديك وكنت تتحدث لغة إنجليزية سليمة فبالتأكيد ستعرف أن هذا الجزء يصف شخص اسمه Judson McDaniel ولد ٢١ فبراير ١٨٣٤ وتوفي عام ١٩٠٥. في الواقع حتى مع الفروق الزمنية أو تلف البيانات يمكن استخلاص أغلب المعلومات ونفس الشيء يمكن أن يقال بالنسبة لبعض جداول البيانات الخاصة أو تنسيق معالج النصوص.

بالإضافة إلى ذلك يعتبر XML موثق بصورة جيدة وتخبرك مواصفات W3C الخاصة بلغة XML 1.0 وكذلك الكثير من الكتب، عن الطرق المثلى لقراءة بيانات XML.

تبادل البيانات فيما بين التطبيقات

نظراً لأن XML ليس ملكية خاصة وسهل في الكتابة والقراءة فانه افضل تنسيق لتبادل ا... فيما بين التطبيقات المختلفة. وإحدى تلك التنسيقات المندرجة تحت التطوير الحالي ... Financial Exchange Format (OFX) وقد صمم هذا التنسيق ليتيح للبرامج المالية الشخصية مثل Microsoft Money و Quicken تبادل البيانات. ويمكن إرسال ... بين البرامج وتبادلها مع البنوك وبيوت السمسرة وغير ذلك.

المرجع إسناد ترافقي: سوف يتم تناول OFX في الفصل ٢.

كما تم الإشارة من قبل فإن XML يعد تنسيق غير مملوك وليس له حقـــوق طبـــع أو بـــراءة اختراع أو سر تجاري أو أي نوع من قيود الملكية العلمية وقد صنع ليصبح ذو فاعلية قصـــوى وليصبح سهل للقراءة والكتابة بالنسبة للأشخاص وبرامج الكمبيوتر ولهذا فهو الاختيار الواضـــح لتبادل اللغات.

باستخدام XML بدلا من تنسيق البيانات الخاصة يمكنك استخدام أي أداة تفهم XML لتعمل مـــع بياناتك ويمكنك حتى استخدام أدوات مختلفة لأسباب عديدة مثلاً برنامج واحـــد للعـــرض وآخـــر للتحرير ويحميك XML من أن تظل حبيس برنامج معين وذلك كون المستند هو المكان المكتـــوب فيه بياناتك أو لأن التنسيق الخاص بالبرنامج هو كل ما يمكن أن يقبله الشخص الذي تراسله.

فمثلاً الكثير من الناشرين يطلبوا استخدام وورد حتى إذا كـــان بإمكـــان المؤلفيـــن اســـتخدام WordPerfect أو Nisus Writer. بذلك يصبح من الصعب على أي شركة أخـــرى أن تقـــوم بنشر معالج للنصوص إلا إذا أمكنهم قراءة وكتابة ملفات وورد ولأن فعل ذلك يتطلـــب مـــبرمج لعكس هندسة تنسيق ملف وورد غير الموثق وهذا يعد استثمار جيد من حيث الوقت والمصـــادر المحدودة المستخدمة. معظم معالجات النصوص الأخرى لها قدرة محدودة لقراءة وكتابة ملفـــات وورد ولكنهم في أغلب الأحيان يفقدوا أثر الرسومات ووحدات المـــاكرو والأنمـــاط وعلامـــات المراجعة وأي ميزات أخرى مهمة. وتكمن المشكلة في أن تنسيق مستند وورد غير موثـــق وذو ملكية خاصة ويتغير بصفة دائمة. ويميل وورد إلى أن ينتهي فائزاً بصورة افتراضية حتـــى وإن فضل الناشرين استخدام برامج أخرى أبسط. إذا تم تطوير تنسيق معالجة نصوص متعارف عليـــه في XML، يستطيع الكتاب استخدام برنامج من اختيارهم.

بيانات منظمة متكاملة

يعد XML الأفضل للمستندات المعقدة الكبيرة وذلك لوضع البيانات في شكل منظم فهو لا يســـمح لك فقط بتحديد المرادفات اللغوية التي تعرف العناصر في المستند ولكن يقوم بتحديد العلاقات بين العناصر أيضاً فمثلا إذا كنت تجمع صفحة ويب لجهات اتصال المبيعات يمكن أن تطلب أن يكون لكل جهة اتصال رقم تليفون وعنوان بريد إلكتروني. إذا كنت تدخل بيانات لقاعدة بيانات يجب أن تتأكد من عدم وجود حقول ناقصة ويمكنك المطالبة بأن يكون لكل كتاب كاتب كما يمكنك تزويـــد قيم افتراضية لتستخدم في حالة عدم وجود بيانات يتم إدخالها.

يزود XML أيضاً جانب العميل بآلية تكمل البيانات من مصادر متعددة وتعرضه كمستند منفرد ويمكن أن يعاد ترتيب البيانات بصورة سريعة وأيضاً يمكن عرض أو إخفاء أجزاء منها بالاعتماد على تصرفات المستخدم وهذا يعد مفيد للغاية عند العمل على كمية ضخمة من المعلومات المخزنة مثل قواعد البيانات العلائقية.

لمحة عن حياة مستند XML

في الأساس XML هو تنسيق مستند، فهو سلسلة من القواعد عن كيف تبدو مستندات XML. هناك مستويان للمطابقة مع معايير XML، الأول هو جودة التكوين والثاني هو التحقق من صحة المستند. الجزء الأول من هذا الكتاب يعرض لك كيف تكتب مستندات جيدة التكوين والجزء الثاني يعرض لك أحسن الطرق التي يمكن بها أن تكتب مستندات سليمة.

تعد HTML "أو لغة ترميز النص المترابط" تنسيق مستند تستخدم على شبكة الإنترنت وداخل مستعرض الويب. يستخدم XML خصيصاً من أجل هذا الغرض ويقوم الكتاب بتوضيح ذلك. ويعتبر XML أكثر قابلية للتطبيق فهو يستخدم، كما أوضحنا من قبل، كتنسيق تخزين لبرامج معالجة النصوص وتنسيق تبادلي لبيانات البرامج المختلفة وكوسيلة للتأكد من التوافق مع قوالب الإنترانت وأيضاً كوسيلة لحفظ البيانات بطريقة يستطيع بها المرء قراءتها.

ومع ذلك فمثل كل تنسيقات البيانات، يحتاج XML إلى برامج ومحتويات قبل أن يصبح فعالاً. فليس مهماً أن تفهم XML نفسه وهو أكثر قليلاً من مجرد مواصفات لما يجب أن يكون عليه شكل البيانات. ستحتاج أيضاً لمعرفة كيف يتم تحرير XML وكيف تقوم المعالجات بقراءة مستندات XML وتمرير المعلومة إلى التطبيقات وما الذي تفعله هذه التطبيقات بالبيانات.

برامج التحرير

في أغلب الأحيان يتم إنشاء مستندات XML بواسطة محرر، وربما سيكون محرر نص أساسي مثل Notepad أو vi والتي في الحقيقة لا تفهم XML على الإطلاق. من جهة أخرى قد يكون محرر WYSIWYG مثل Adobe FrameMaker الذي يعزلك تماماً عن تفاصيل تنسيق XML أو ربما يكون محرر هيكلي مثل JUMBO الذي يعرض مستندات XML كشجرة. وعامة لن تعطيك أغلب برامج التحرير النتيجة التي تريدها لذا يركز هذا الكتاب على كتابة XML يدوياً في محرر نصوص.

يمكن للبرامج الأخرى إنشاء مستندات XML فمثلا في هذا الكتاب وبالتحديد في الفصل الذي يتناول تصميم DTD جديد سترى بعض بيانات تأتي رأسا من قاعدة بيانات FileMaker. في هذه الحالة يتم إدخال البيانات أولا في قاعدة بيانات FileMaker ثم يقوم حقل حسابات FileMaker بتحويل البيانات إلى XML وعامة يعمل XML بشكل رائع مع قواعد البيانات.

المرجع ستری هذا الجزء بالتحديد في الفصل ٢٣ "تطبيق تصميم XML جديد".

على كل حال فإن المحرر أو أي برنامج آخر هو الذي يقوم بإنشاء مستندات XML. وفي أغلب الأحيان إن لم يكن كلها فإن هذا المستند هو ملف فعلي على القرص الثابت للكمبيوتر، فمثلا المستند ربما يكون سجل أو حقل في قاعدة بيانات أو ربما كان سلسلة من البايتات تم استقبالها من شبكة الاتصال.

الموزعات والمعالجات

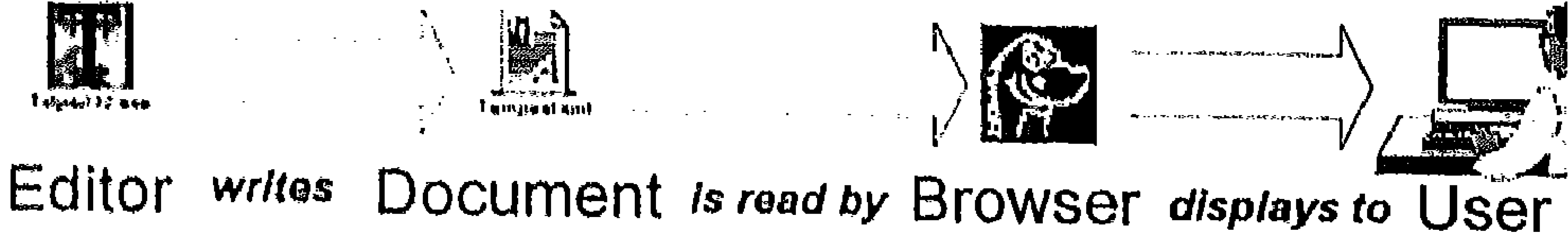
يقرأ موزع XML والمعروف أيضاً باسم معالج XML المستند ويتحقق من أن XML الموجود بداخله تم تصميمه جيداً ويتحقق من صحة المستند على الرغم من عدم وجوب هذا الاختبار. يتم تناول تفاصيل هذه الاختبارات في الجزء الثاني. ولكن مع افتراض أن المستند يقوم بإجراء هذه الاختبارات فإن المعالج يقوم بتحويل المستند إلى شجرة من العناصر.

المستعرض وأدوات أخرى

وأخيراً يقوم المعالج بتمرير الشجرة أو الفروع المنفردة للشجرة إلى التطبيق النهائي. ربما يكون هذا التطبيق مستعرض مثل Mozilla أو أي برنامج آخر يفهم ما الذي يجب فعله مع البيانات. إذا كان هذا البرنامج مستعرض فإن البيانات سيتم عرضها للمستخدم ولكن البرامج الأخرى ربما تستقبل البيانات أيضاً. فمثلا ربما تفهم البيانات كإدخال إلى قاعدة البيانات أو سلسلة من العلامات الموسيقية التي يتم عزفها أو برنامج Java الذي يجب تشغيله ويعتبر XML مرن جداً ويمكن استخدامه في العديد من الأغراض.

تلخيص عملية المعالجة

ليتم تلخيص ما سبق فإن مستند XML يتم إنشاءه في محرر ويقوم المعالج بقراءة المستند وتحويله إلى شجرة من العناصر وكذلك يمرر المعالج الشجرة إلى المستعرض الذي يعرضه. يوضح الشكل ١-١ هذه العملية.



الشكل ١-١ دورة حياة مستند XML

من الضروري ملاحظة إن كل هذه الأجزاء مستقلة وغير مرتبطة ببعضها و الشيء الوحيد الذي يربطهم كلهم معاً هو مستند XML. يمكن تغيير محرر البرامج فقط دون تغيير التطبيق النهائي وفي الواقع قد لا تعرف ما هو التطبيق النهائي ربما يكون مستخدم يقرأ عملك أو ربما تكون قاعدة بيانات تستخدمها أو ربما تكون شئ لم يخترع بعد أو كل ذلك معا ويكون المستند مستقل عن البرامج التي تقرأ المستند.



تعد لغة HTML مستقلة عن البرامج التي تستطيع قراءتها وكتابتها ولكنها مناسبة فقط للاستعراض. وتعتبر بعض الاستخدامات الأخرى مثل إدخالات قاعدة البيانات خارج نطاقها. على سبيل المثال لا تزودك لغة HTML بوسيلة تفرض على الناشر تضمين محتوى مطلوب محدد مثل طلب أن يكون لكل كتاب رقم ISBN أما في XML يمكن طلب تنفيذ ذلك ويمكن كذلك فرض الترتيب الذي تظهر به بعض العناصر المحددة. على سبيل المثال رؤوس أعمدة المستوى الثاني يجب أن تتبع رؤوس أعمدة المستوى الأول.

تقنيات مرتبطة

استخدام XML فيما يزيد عن كونه تنسيق بيانات يتطلب تفاعل مع عدد من التقنيات المرتبطة والتي تتضمن HTML للتوافق مع النسخ الأقدم من المستعرضات الموجودة ولغات أوراق نمط CSS وURLs وURIs ولغات ربط XLL و مجموعة حروف Unicode.

لغة ترميز النص المترابط

يعد إصدار Mozilla 5.0 وInternet Explorer 5.0 من أول مستعرضات الويب التي تقدم بعض الدعم للغة XML ولكنها تأخذ فترة عامين قبل أن يقوم أغلب المستخدمين بتحديث أجهزتهم إلى نسخ معينة من البرامج الجديدة لذا ستحتاج إلى تحويل محتوى XML إلى HTML لبعض الوقت.

لذا قبل أن تقوم بتشغيل XML يجب أن تكون على دراية بلغة HTML. لن تحتاج لأن تكون مصمم محترف ولكن يجب أن تعرف الرابطة بين الصفحة والتي تليها وكيفية تضمين صورة في المستند وكيف تجعل النص بالخط العريض وبما أن HTML يعتبر أكثر تنسيقات مخرجات XML شيوعا ولذلك كلما كان استخدام لغة HTML أسهل كلما كان من السهل إنشاء التأثير الذي تريده.

على الصعيد الآخر، إذا كنت معتادا على استخدام الجداول أو صور GIF ذات البكسل المنفرد لترتيب الكائنات على الصفحة أو إذا بدأت في إنشاء موقع ويب بتصميم مظهره بدلا من محتواه إذن يجب أن تنتهي من ممارسة بعض العادات السيئة. كما شرحنا من قبل يقوم XML بفصل محتوى المستند عن مظهره. يتطور المحتوى أولا ثم يتم إرفاق تنسيق بهذا المحتوى عن طريق ورقة نمط. يعتبر فصل المحتوى عن النمط تقنية فعالة جدا تحسن المحتوى ومظهر المستند وضمن أشياء أخرى تسمح للمحررين والمصممين بالعمل بصورة أكثر استقلالية ومع ذلك فأنها تتطلب طرق تفكير مختلفة بشأن تصميم موقع ويب وأيضاً استخدام تقنيات إدارة مشروعات جديدة عند وجود العديد من الناس.

أوراق النمط

حيث أن لغة XML تسمح بوجود علامات خاصة بالمستخدم لإضافتها في مستند ولا توجد طريقة ليعرف بها المستعرض مسبقاً كيف يتم عرض كل عنصر. لذلك فعندما ترسل مستند إلى مستخدم تحتاج أيضاً إلى إرسال ورقة نمط التي تطلع المستعرض عن كيفية تنسيق كل عنصر على حدة وإحدى أنواع أوراق النمط الممكن استخدامها هي Cascading Style Sheets ويرمز لها بالحروف CSS "أوراق النمط".

يعرف (CSS)، الذي صمم في البداية من أجل HTML، خصائص التنسيق مثل حجم ونوع وشكل الخط وشكل الفقرة ومحاذاتها وأنماط أخرى يمكن تطبيقها على عناصر محددة. فمثلا يسمح CSS لمستندات HTML بتحديد أن كل عناصر H1 يجب تنسيقها باستخدام حجم ٣٢ نقطة بخط Helvetica أسود عريض في المنتصف. يمكن تطبيق الأنماط المنفردة على أغلب علامات HTML التي تتجاوز عن افتراضات المستعرض ويمكن تطبيق أوراق أنماط متعددة على مستند منفرد ويمكن تطبيق أنماط متعددة على عنصر منفرد وتتالى الأنماط تبعاً طبقاً لمجموعة معينة من القواعد.

المرجع

تم تناول قواعد CSS وخصائصها بالتفصيل في الفصل ١٢ "أوراق نمط المستوى١" والفصل ١٣ "أوراق نمط المستوى٢".

من السهل تطبيق قواعد CSS على مستندات XML ويمكنك ببساطة تغيير أسماء العلامات التي يتم تطبيق القواعد عليها ويقوم Mozilla 5.0 بتدعيم أوراق نمط CSS المرتبطة بمستندات XML.

لغة النمط القابلة للمد

تعتبر لغة (XSL) هي أكثر لغة أوراق النمط تقدماً والتي صممت خصيصاً لتستخدم مع مستندات XML وتعد مستندات XSL مستندات XML جيدة التكوين.

تحتوي مستندات XSL على سلسلة من القواعد التي تطبق على أنماط معينة من عناصر XSL. يقرأ معالج XSL مستندات XML ويقارن ما يراه بالأنماط في ورقة النمط وعندما يتم ملاحظة نمط من ورقة نمط XSL في مستندات XML تقوم القاعدة بوضع بعض التركيبات النصية وعلى عكس أوراق النمط المتتالية فإن مخرجات هذا النص تكون عشوائية ولا تكون محددة بالنص الذي تم إدخاله ومعلومات التنسيق.

تستطيع CSS تغيير تنسيق عنصر محدد فقط وتفعل ذلك على أساس عنصر فقط، وعلى صعيد آخر يمكن لأوراق نمط XSL إعادة ترتيب وتنظيم العناصر حيث بإمكانها إخفاء بعض العناصر وعرض غيرها والأكثر من ذلك بإمكانها اختيار النمط المستخدم ليس فقط على أساس العلامات ولكن على أساس المحتويات وسمات العلامات وعلى أساس موقع العلامة في المستند بالنسبة إلى العناصر الأخرى وبناء على عدة معايير أخرى.

لدى CSS ميزة دعم أكبر للمستعرضات ولكن XSL أكثر مرونة وفاعلية والأفضل لمستندات XML والأكثر من ذلك أن مستندات XML مع أوراق نمط XSL من السهل تحويلها إلى مستندات HTML مع أوراق نمط CSS.

المرجع

يتم شرح أوراق نمط XSL بالتفصيل في الفصل ١٤ "لغة تحويل XSL" والفصل ١٥ "كائنات تنسيق XML".

URIs و URLs

بإمكان مستندات XML البقاء على الويب تماماً مثل HTML والمستندات الأخرى وعندما تفعل ذلك يشار إليهم بواسطة محددات المواقع تماماً كملفات HTML. فمثلاً في موقع http://www.hypermedic.com/style/xml/tempest.xml ستجد النص الكامل لمسرحية الكاتب المسرحي شكسبير Tempest التي تم ترميزها باستخدام لغة XML.

بالرغم من أن URL مفهوم جيداً ومدعم جيداً فـــأن مواصفـــات XML تســـتخدم Uniform Resource Identifier (URIs) الأكثر عمومية وهو بناء لتحديد موقع المصادر على الإنـــترنت والذي يركز بصورة كبيرة على المصدر وبصورة أقل على الموقع. نظرياً، يستطيع URI أن يجد أقرب نسخة من المستند أو يعين موقع المستند الذي تم نقله من موقع إلى آخـــر ومـــن الناحيـــة العملية مازالت URLs مجال نشط للبحث والأنواع الوحيدة من URLs التـــي تدعمـــها البرامـــج الحالية هي URLs.

Xlinks و Xpointers

طالما أن مستندات XML يتم إرسالها على الإنترنت فأنك تحتاج إلى إمكانية وضع عنـــاوين لـــها وأن يكون هناك رابطة بينها. وعلامات ربط HTML القياسية يمكن استخدامها في مستندات XML ومستندات HTML يمكن ربطها بمستندات XML فمثلا تشير رابطة HTML بالإشارة إلى نســـخة Tempest الموجودة في XML السابقة الذكر.

```
<a href="http://www.hypermedic.com/style/xml/tempest.xml">
 The Tempest by Shakespeare
</a>
```



يتوقف ما إذا كان المستعرض يستطيع عرض هذا المستند إذا تتبعت الرابطة على كيف يتعامل المستعرض مع ملفات XML وأغلب المستعرضات الحالية لا تستطيع معالجتهم بصورة جيدة.

ومع ذلك، يتيح XML التعامل مع Xlinks للربط بين المستندات وXPointers لتعريف أجـزاء منفصلة من المستند.

تتيح Xlinks لأي عنصر أن يصبح رابطة وليس فقط عنصر A. والأكثر مـــن ذلـــك يمكـــن للارتباطات أن تكون ثنائية أو متعددة الاتجاه أو حتى يشير إلي أقرب نسخة من الموقع حيث يتـــم تحديد أقرب الروابط وتستخدم Xlinks URLs العادية لتحديد الموقع الذي يتم الربط إليه.

المرجع يتناول الفصل ١٦ "Xlinks".

تتيح XPointers للروابط الإشارة ليس فقط لمستند محدد في موقع محدد ولكن إلـــى جـــزء محدد في مستند معين. بإمكان XPointers الإشارة إلى عنصر محدد من المستند إلـــى الأول أو

الثاني أو السابع عشر والى العنصر الأول الذي يعد تابع لعنصر موجود وهكذا، وتقدم XPointers روابط قوية بين المستندات التي لا تتطلب أن يحتوي المستند الهدف على ترميز إضافي حتى يتم ربط الأجزاء المنفردة إليه.

وأكثر من ذلك فعلى عكس ارتساء HTML فأن Xpointers لا تشير فقط إلى نقطة في مستند حيث بإمكانها الإشارة إلى نطاقات وفترات زمنية ولذا يمكن استخدام XPointers لتحديد جزء محدد من المستند وذلك ربما لنسخه أو تحميله في برنامج.

المرجع سوف يتم شرح XPointers في الفصل ١٧.

مجموعة حروف Unicode

تعد شبكة ويب شبكة دولية ومع ذلك ستجد إن أغلب النصوص موجودة باللغة الإنجليزية ولكن XML بدأ في تغيير ذلك. يقوم XML بتقديم الدعم الكامل لمجموعة حروف Unicode ثنائية البايت وكذلك تمثيلها المصغر وتدعم مجموعة الأحرف هذه تقريباً كل الحروف الشائعة الاستخدام في جميع النصوص الحديثة في العالم.

لسوء الحظ فأن XML وحده ليس كافيا. لقراءة نص تحتاج إلي ٣ أشياء:

١- مجموعة أحرف للنص.

٢- نوع خط لمجموعة الأحرف.

٣- نظام تشغيل وتطبيق يتفهم مجموعة الأحرف.

إذا أردت الكتابة في نص بالإضافة إلى قراءته ستحتاج إلى طريقة إدخال للنص ومع ذلك فإن XML يعرف مراجع أحرف تتيح استخدام ASCII لتشفير الأحرف غير المتاحة في مجموعة الأحرف الأصلية. يعتبر ما سبق كافياً عند الاقتباس من لغات مثل اليونانية والصينية على الرغم من عدم إمكانية الاعتماد عليه في كتابة رواية بلغة أخرى.

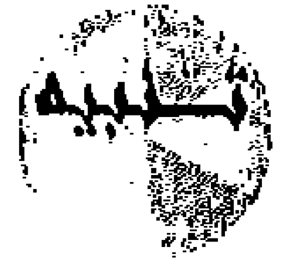
المرجع في الفصل ٧ ستتعرف على المزيد عن اللغات المختلفة والنصوص الدولية الممثلة في الكمبيوتر وكيف يفهم XML النص وكيف يمكنك استخدام البرنامج الموجود لقراءة وكتابته بلغات أخرى غير اللغة الإنجليزية.

كيف تلائم التقنيات بعضها البعض

يقوم XML بتحديد قواعد اللغة للعلامات التي تستخدمها لترميز مستند. يتم ترميز مستندات XML بعلامات XML والتشفير الافتراضي لمستندات XML هو Unicode.

ضمن أشياء أخرى ربما يحتوي مستند XML على روابط نص مترابط لمستندات أو مصادر أخرى وقد تم إنشاء هذه الروابط تبعاً لمواصفات Xlinks. يقوم Xlinks بتحديد المستندات التي تم الربط إليها مع URIs نظرياً أو URLs عملياً وربما يحدد Xlinks الجزء المنفرد من المستند المرتبط به وهذه الأجزاء يتم التعامل معها عبر Xpointers.

إذا كان من المقدر قراءة مستند XML بواسطة بعض الناس وليس كل مستندات XML فأن ورقة نمط تقوم بتقديم تعليمات عن كيفية تنسيق العناصر المنفصلة ويمكن أن تكتب ورقة النمط في أي من لغات أوراق النمط العديدة ويعد CSS وXSL هم أكثر لغات أوراق النمط شيوعاً على الرغم من وجود لغات أخرى تتضمن DSSCL وهي اختصار لجملة Document Style Semantics and Specification Language التي تعتمد عليها XSL.



لقد كتبت الكثير في هذا الفصل إلا أن الأمانة تحتم علي أن أخبرك إنني لم أتناول كل شئ بعد بل الحقيقة هو أني قد تناولت المتوقع من XML وليس الواقع ولكن يوجد الكثير من المبرمجين الذين يسعون لجعل الحلم حقيقة فهناك برامج جديدة يتم إخراجها كل يوم لنقترب أكثر من XML ولكن ما زال هناك الكثير في مرحلة الإعداد. في بقية هذا الكتاب سأكون حذرا في الإشارة ليس فقط إلى المتوقع ولكن إلى ما يحدث فعلاً ولكن للأسف في أغلب الأحيان فأن الاثنان ليسوا نفس الشيء ولكن مع القليل من الحذر يمكنك الآن إنجاز عمل حقيقي باستخدام XML.

خلاصة

لقد تعلمت في هذا الفصل بعض الأشياء التي بإمكان XML إنجازها لك مثل:

- XML هي لغة ترميز معلومات تمكنك من إنشاء لغات ترميز لمستندات ومجالات معينة.
- علامات XML تصف بنية ومحتوى المستند وليس تنسيق المحتوى. يتم وصف التنسيق في ورقة نمط منفصلة.
- يتم استخدام XML لتجنب صعوبة التعامل SGML وHTML.
- يتم إنشاء XML في محرر يقرأ بواسطة معالج ويعرض عن طريق مستعرض.

- يظل XML الموجود على الويب على الأساس المزود من قبـــل HTML وأوراق النمــط المتتالية وURL.
- هناك العديد من تقنيات التدعيم تعتمد على XML وتتضمـــن أوراق نمــط XSL و xlinks و Xpointers وهؤلاء يسمحوا لك بأكثر مما يمكنك إنجازه بواسطة CSS و URLs.
- كن حذراً فإن XML لم ينتهي بعد، فإنه سيتغير ويتسع وربما تقابل مشاكل فــــي برامــج XML الحالية.

في الفصل التالي ترى عدد من تطبيقات XML وتتعلم بعض الطرق التي يستخدمها XML فــي عالمنا اليوم وتتضمن الأمثلة الرسوم الاتجاهية والرياضيـــات والكيميـــاء والمصـــادر البشــرية وغيرها.



الفصل

مقدمة إلى تطبيقات XML

في هذا الفصل

يستعرض هذا الفصل بعض الأمثلة من تطبيقات XML ولغات الترميز المستخدمة لإضافة العديد من العناصر التي تحسن XML ووظائفها التي قد تكون غير واضحة. ويفضل معرفة بعض الاستخدامات التي من أجلها تم وضع XML حتى في هذه المرحلة الأولى منه، كما يقدم هذا الفصل فكرة عن تطبيقات XML بيد أنه لا يخفى على القارئ أنه جاري إنشاء المزيد من التطبيقات التي تعتمد على XML في أثناء كتابة هذا الكتاب.

المرجع يغطي الجزء الخامس التطبيقات التي تمت مناقشتها في هذا الفصل على نحو مفصل.

ما هو تطبيق XML

تعد XML لغة ترميز تفصيلية لتصميم لغات ترميز محددة وتسمي كل لغة ترميز تستخدم XML تطبيق XML غير أن هذا ليس تطبيقاً يستخدم مستعرض مثل Mozilla Web وجدول بيانات Gnumeric أو محرر XML Pro ولكنه يعتبر تطبيق من XML موجه إلى مجال محدد مثل Chemical Markup Language أو لغة الترميز الكيميائية والتي يرمز لها بالرمز (CML) وGedML لعلم الأنساب.

يحتوي كل تطبيق من تطبيقات XML على بناء جملة ومعجم خاص به. ويقترب بناء الجملة هذا والمعجم من القواعد الأساسية في XML التي هي تقترب بدورها من لغات البشر التي يكون لكل منها معجمها وقواعدها النحوية الخاصة بها وفي نفس الوقت تقترب من قوانين أساسية فرضتها طبيعة تشريح الجسد الإنساني وبناء المخ.

يعد XML تنسيق مرن للغاية للبيانات التي تعتمد على النص والسبب وراء اختيار XML كأساس للتطبيقات المختلفة التي يتم تناولها في هذا الفصل هو أن XML يزود المستخدم بتنسيق محكم الصياغة يسهل قراءته وكتابته. حيث أنه يمكن باستخدامها التعامل مع كمية هائلة من البيانات عن طريق تحويل مجموعة من المعالجات المفصلة إلى الأدوات ومكتبات إضافية، والأكثر من ذلك أنه من سهولة وضع مستويات إضافية من بناء الجمل وعلم الألفاظ بالإضافة إلى البناء الأساسي الذي يقدمه XML.

الترميز الكيميائية

تعتبر لغة الترميز الكيميائية والتي يرمز لها بالرمز CML من أول تطبيقات XML. وقد صمم هذا التطبيق كتطبيق SGML ثم تم نقلها بالتدريج إلى XML. تعتبر CML هي HTML في أغلب الأحوال غير أنها تحتوي على بعض الجزئيات الإضافية كما أنها تحتوي على تطبيقات أكثر من التطبيقات المحددة والمرتبطة بالويب.

تحتوي المستندات الجزئية على الآلاف من الكائنات المختلفة شديدة التفصيل، فمثـــلا ربمـا يحتوي جزيء واحد متوسط الحجم على مئات من الذرات كل منها له العديد من الروابط. تسـعى CML إلى تنظيم هذه الكائنات الكيميائية المعقدة بطريقة منتظمة يمكن فهمها وعرضها والبحـــث عنها باستخدام الحاسب ويمكن استخدام CML لبنية الجزيئات والتتابعات وتحليل الرسوم البيانيـــة للطيف وعلم البلوريات والنشر وقواعد البيانات الكيميائية وغير ذلك. يتضمن المعجـم جزيئـات وذرات وروابط وبلورات وصيغ وتتبعات وردود الأفعال ومصطلحات كيميائيـة أخـرى فمثــلا تعليمات البرمجة ٢-١ ما هي إلا مستند CML لجزئ الماء (H2O):

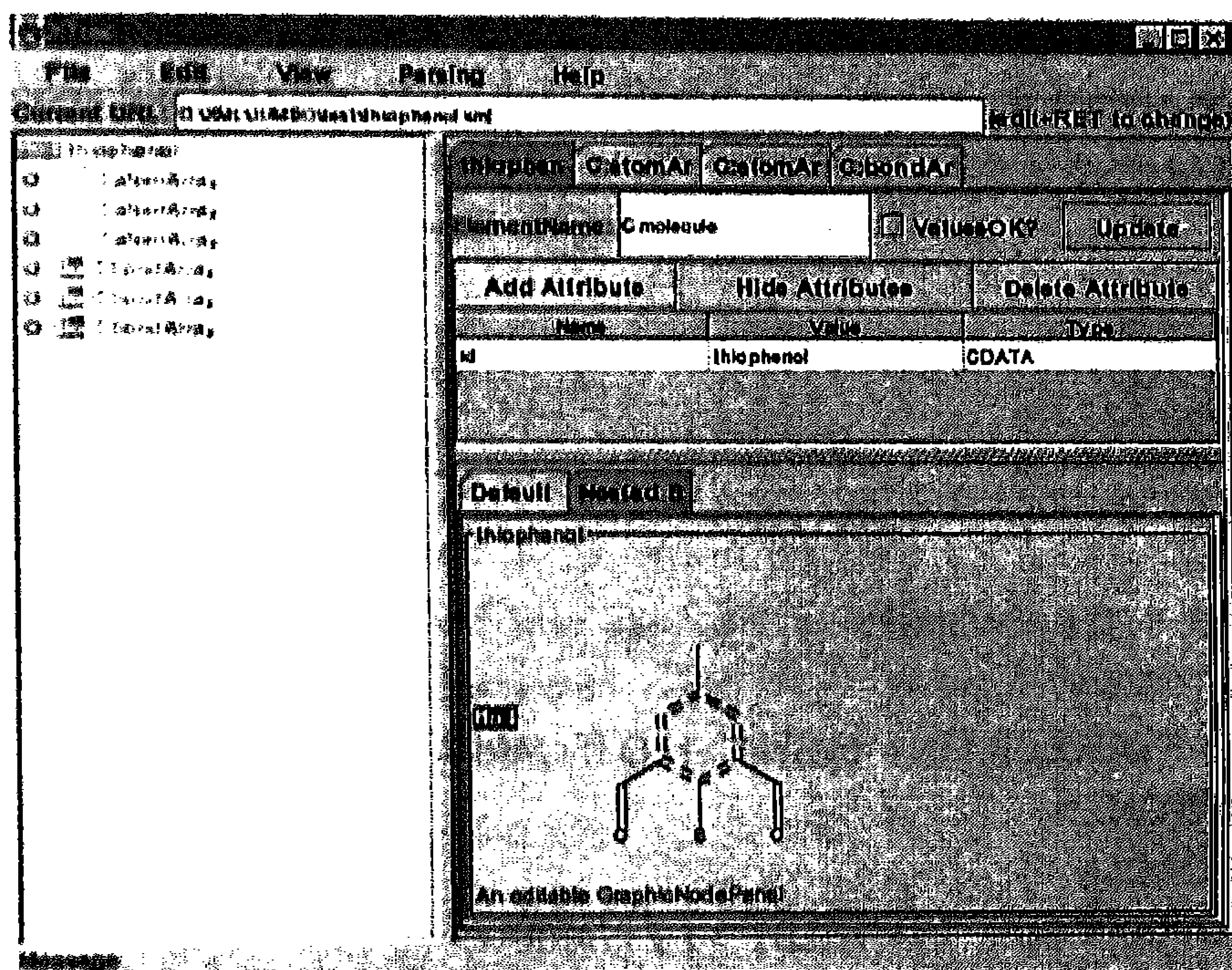
تعليمات البرمجة ٢-١: جزئ المياه H2O

```
< ? xml  version="1.0"?>
<CML>
  <MOL TITLE="Water">
    <ATOMS>
      <ARRAY BUILTIN="ELSYM">H O H</ARRAY>
    </ATOMS>
    <BONDS>
      <ARRAY BUILTIN="ATID1">1 2</ARRAY>
      <ARRAY BUILTIN="ATID2">2 3</ARRAY>
      <ARRAY BUILTIN="ORDER">1 1</ARRAY>
    </BONDS>
  </MOL>
</CML>
```

تعد سهولة البحث من أفضل الميزات التي تقدمها CML لإدارة البيانات الكيميائية. كما تتيـــح CML أيضاً إرسال بيانات الجزئيات المعقدة عبر الويب. ولأن XML المضمنة هي نظام تشـــغيل مستقل فإنها تجنب المستخدم مشكلة استقلالية نظام التشغيل التي تظهر فـــي التنسـيقات الثنائيـــة المستخدمة بواسطة البرامج الكيمائية والمستندات مثل تنسـيقات Protein Data Bank والتـــي يرمز لها PDB وMDL Molfiles.

قام موراي رست بعمل أول مستعرض XML يتم استخدامه لأغراض عامة وقد أطلق عليـــه Jumbo. ويوضح الشكل ٢-١ ملف CML يعرضه Jumbo، وهو يعمل عن طريق تعيين كـــل عنصر XML إلى فئة Java التي تعرف كيفية تقديم هذا العنصر. يمكن ببساطة كتابة فئات Java

لعناصر جديدة لتسمح Jumbo بدعمها. يتم توزيع Jumbo مع فئات لعرض المجموعة الأساسية من عناصر CML متضمنة جزئيات وذرات وهي متاحة على العنوان التالي .http://www.xml-cml.org/.



الشكل ٢-١ مستعرض Jumbo يعرض ملف CML.

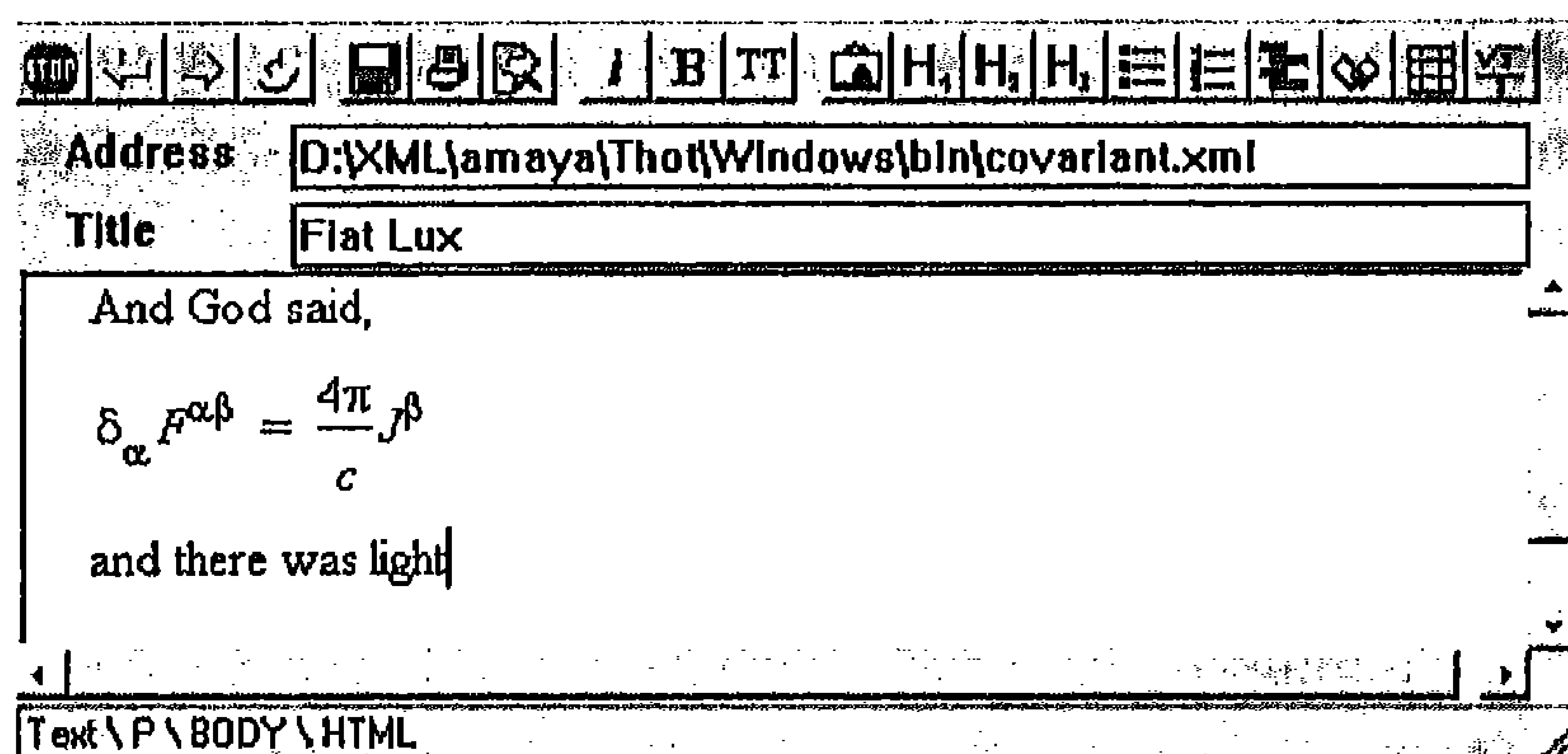
لغة الترميز الحسابية

يذكر البعض أن تيم برنرز لي قد أنشأ شبكة الويب العالمية وHTML في CERN مما أتاح لعلماء الطبيعة تبادل الأبحاث والمواد العلمية التي ما زالت تحت التجربة غير أن هذا الاعتقاد غير صحيح لأن العنصر المشترك بين كل من علم الطبيعة والفيزياء والرياضيات التطبيقية وعلم الفضاء وعلوم الحاسب الآلي هو وجود الكثير والكثير من المعادلات. وحتى الآن وبعد تسع سنوات من اختراع الويب لا توجد طريقة جيدة لتضمين المعادلات في صفحات الويب.

تظهر بعض الملامح في تطبيقات Java الصغيرة التي تقوم بمعالجة بناء الجمل والمحولات التي تحول مجالات LaTeX إلى صور GIF والمستعرضات المخصصة التي تستطيع عرض ملفات TeX ولكن لا ينتج عنها نتائج متميزة كما أنها لم تكن محط إعجاب من محرري صفحات الويب. وهنا تكمن مميزات هذه اللغة حيث أنها بداية التغيير لكل هذا.

تعد لغة الترميز الرياضية والتي يرمز لـــها بـــالرمز MathML تطبيـــق XML للمعـــادلات الرياضية وهي كافية للتعامل مع كل النماذج الرياضية بجميع مستوياتها تقريباً كما أنه بإمكانـــها التعامل مع مواضيع متقدمة لذلك على الرغم من وجود بعض العيوب في بعض العلامات الغريبة المتقدمة المستخدمة في بعض أقسام علم الرياضيات. وعلى الرغم من العيوب التي تظهر في هذه اللغة فإنه بإمكان MathML التعامل مع أغلـــب الاحتياجـــات التعليميـــة والعمليـــة والهندســـية والاقتصادية والإحصائية. ستتسع خدمات MathML بحيث يتمكن كل الرياضيين والباحثين مـــن النشر والبحث على الويب. ستحول هذه اللغة الويب إلى أداة مهمة للبحث العلمي والاتصالات.

مازال Netscape Navigator وInternet Explorer لا تدعـــم MathML. وقـــد قـــامت W3C بدمج بعض دعم MathML في مستعرض الاختيار ويعـــرض الشـــكل ٢-٢ مســـتعرض Amaya وهو يعرض نموذج لمعادلات Maxwell المكتوبة باستخدام MathML.



الشكل ٢-٢ يعرض مســـتعرض Amaya نمـــوذج لمعـــادلات MathML المكتوبة بلغة MathML.

ملف XML الذي يعرضه مستعرض Amaya مكتوب فيما يلي:

تعليمات البرمجة ٢-٢ : معادلات Maxwell في MathML

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/TR/REC-html40"
    xmlns:m="http://www.w3.org/TR/REC-MathML/"
>
<head>
<title>Fiat Lux</title>
```

```
<meta name="GENERATOR" content="amaya V1.3b" />
</head>
<body>

<P>
And God said,
</P>

<math>
  <m:mrow>
    <m:msub>
      <m:mi>&delta;</m:mi>
      <m:mi>&alpha;</m:mi>
    </m:msub>
    <m:msup>
      <m:mi>F</m:mi>
      <m:mi>&alpha;&beta;</m:mi>
    </m:msup>
    <m:mi></m:mi>
    <m:mo>=</m:mo>
    <m:mi></m:mi>
    <m:mfrac>
      <m:mrow>
        <m:mn>4</m:mn>
        <m:mi>&pi;</m:mi>
      </m:mrow>
      <m:mi>c</m:mi>
    </m:mfrac>
    <m:mi></m:mi>
    <m:msup>
      <m:mi>J</m:mi>
    <m:mrow>
      <m:mi>&beta;</m:mi>
      <m:mo></m:mo>
```

```
      </m:mrow>
    </m:msup>
  </m:mrow>
</math>

<P>
and there was light
</P>
</body>
</html>
```

تقدم تعليمات البرمجة ٢-٢ مثالاً على نموذج صفحة تم إنشائها باستخدام كـــل مـــن HTML وXML. تم كتابة عناوين الصفحات وفقرات النص "fiatLux وMaxwell's Equations و And God said و"and there Was light باستخدام HTML التقليـــدي والمعـــادلات الفعليـــة فـــي MathML وهو من أحد تطبيقات XML.

تحتاج مثل هذه الصفحات المختلطة "أي التي تمت كتابتها باستخدام كل مـــن HTML و"XML إلى دعم خاص من المستعرض كما أنها تحتاج إلى إضافات أو برنامج JavaScript أو عنـــاصر تحكم ActiveX التي تعالج بيانات XML المضمنة وتعرضها، كما تحتاج بالطبع إلى مســـتعرض مثل Mozilla 5.0 أوInternet Explorer الذي بإمكانه معالجة وعرض ملفات XML الخالصة بدون تدخل HTML.

خصائص تعريف القنوات

يعد Channel Definition Format والذي يرمز له بالرمز (CDF) من أحـــد تطبيقـــات XML لتعريف القنوات ويستخدم موقع الويب القنوات لتحميل المعلومات للقراء المشتركين في الموقـــع بدلا من الانتظار للحصول عليها وقد أطلق على ذلك Webcasting أو Push. تـــم تقديـــمCDF لأول مرة في Internet Explorer 4.0.

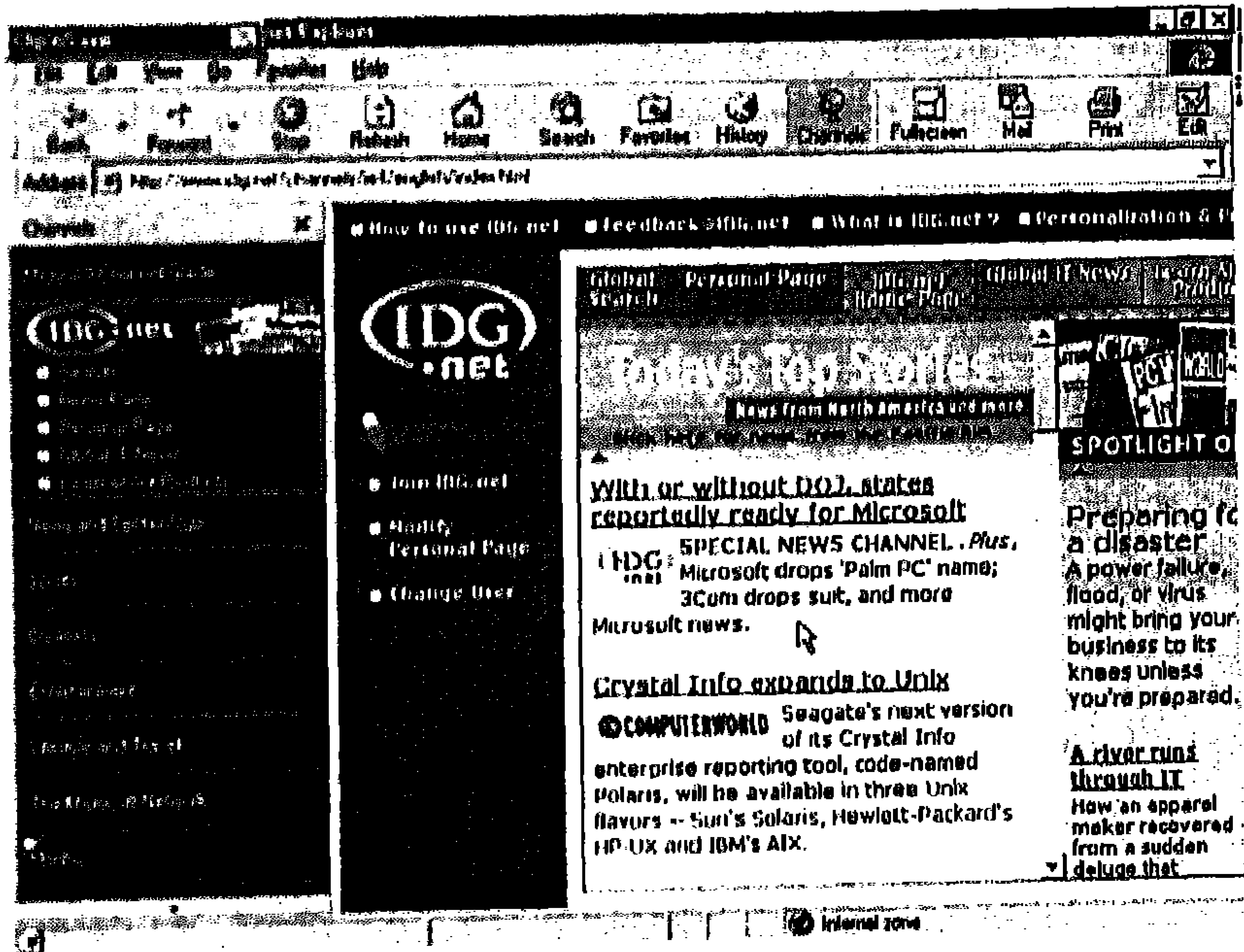
يعد مستند CDF ملف XML مفصل عن مستند HTML لكنه يرتبط به في الموقع. تحدد القنـــاة في مستند CDF الصفحات التي يتم إرسالها آلياً للقراء وكيف يتم نقل هذه الصفحات وعدد المرات التي يتم إرسالها، ومن الممكن تطبيق تقنية Webcasting أو Push عن طريق إرسال بعـــض العينات أو حتى مواقع ويب كاملة إلى المشتركين.

يمكن إضافة CDF إلى موقعك دون تغيير محتواه ويمكن إضافة ارتباط غير مرئي إلى ملف CDF على الصفحة الأساسية، وعند زيارة أحد القراء الصفحة فأن المستعرض يعرض مربع حوار يستفسر عما إذا كان يريد الاشتراك في القناة، وإذا أبدى القارئ رغبته في ذلك فإن المستعرض يربط المعاملات المحددة في مستند CDF مع ما يفضله المستخدم من خيارات خاصة لتحديد وقت الاتصال مع الخادم للاطلاع على المحتوى الجديد. وفي واقع الأمر لا يعد ذلك Push بكامل معنى هذه التقنية حيث أن العميل يجب عليه بدء الاتصال وهو الأمر الذي يحدث بدون طلب مباشر من القارئ. يعرض الشكل ٢-٣ IDG Active Channel في Internet Explorer 4.0.

المرجع يتم تغطية CDF بالتفصيل في الفصل ٢١ "الدفع بمواقع ويب عن طريق CDF".

الأدب القديم

قام جون بوساك بكتابة الأعمال الكاملة لشكسبير باستخدام صفحات XML كما أنه قد قام بتضمين النصوص الكاملة مع الأعمال المكتوبة باستخدام ترميز XML للتفرقة بين العناوين والعناوين الفرعية والحركة والأسطر والمستخدمين، وما إلى ذلك.



الشكل ٢-٣ IDG Active Channel في Internet Explorer 4 0

وقد يتساءل القارئ عن أهمية ذلك، إلا أن أهميته تتضح عندما يتم التعامل معه من قبل الحاسب الآلي الذي يتميز بالقدرة على تحليل النصوص فيقدم إمكانية التفرقة بين العناصر المختلفة التي تم كتابة المسرحيات على أساسها. فمثلا من السهل أن يقوم بالعثور على كل الأسطر التي وردت على لسان روميو.

والأكثر من ذلك أنه عند تبديل ورقة النمط التي تم بها تنسيق المستند فإنه بإمكان الممثل أن يطبع أحد النسخ التي يظهر فيها الدور الخاص به بالخط السميك مع جعل كلمات الأدوار التي تليها مائلة. يمكن فصل نصوص معينة قام بأدائها شخص معين في المسرحية باستخدام XML وبطريقة أسهل بكثير منها في حالة كتابة المسرحية بالطريقة العادية.

وقد قام بوساك بكتابة ترجمات إنجليزية من العهد القديم والعهد الجديد والقرآن باستخدام XML. تختلف عملية الترميز في هذه النماذج عن غيرها فمثلا لا يمكن التفرقة بين المتحدثين لذا لا يمكن استخدام مستندات XML لإنشاء إنجيل مكتوب بالخط الأحمر "أي الذي تظهر فيه الكلمات التي قالها السيد المسيح باللون الأحمر" على الرغم من إمكانية عمل ذلك باستخدام مجموعة مختلفة من العلامات. ولأن أغلب هذه الملفات تكتب بالإنجليزية بدلا من اللغات الأصلية فإنها غير مفيدة للتحليل النصي من الناحية الدراسية وسوف تحتاج إلى وضع بعض الكلمات الأخرى غير التي وضعها بوساك لوصف نفس العناصر التي أراد أن يصفها.

لغة تكامل الوسائط المتعددة المتزامنة

تعد Synchronized Multimedia Integration Language "وتختصر "SMIL تطبيق من تطبيقات XML وW3C لكتابة أشكال للوسائط المتعددة بشكل تليفزيوني لا تقوم مستندات SMIL بوصف محتوى الوسائط المتعددة "وهي الفيديو أو الصوت الذي يتم تشغيله" ولكن يختص بوقت ومكان التشغيل.

فمثلا مستند SMIL لاحتفال مصور ربما يطلب من المستعرض تشغيل ملف الصوت beethoven9.mid وعرض ملف الفيديو corange.mov وعرض ملف HTML المسمى clockwork.htm ثم عند الانتهاء من ذلك يجب أن يقوم المستعرض بعرض ملف فيديو 2001.mov وملف الصوت zarathustra.mid وعرض ملف HTML المسمى aclarke.htm. مما يؤدي إلى عدم الحاجة إلى تضمين نوع بسيط من البيانات مثل النص في نوع معقد من البيانات مثل الفيديو. وتوضح تعليمات البرمجة ٢-٣ أحد ملفات SMIL التي توضح ذلك.

تعليمات البرمجة ٢-٣: A SMIL film festival

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
  "http://www.w3.org/TR/REC-smil/SMIL10.dtd">
<smil>
  <body>
    <seq id="Kubrick">
      <audio src="beethoven9.mid"/>
      <video src="corange.mov"/>
      <text src="clockwork.htm"/>
      <audio src="zarathustra.mid"/>
      <video src="2001mov"/>
      <text src="aclarke.htm"/>
    </seq>
  </body>
</smil>
```

بالإضافة إلى تحديد زمن تتابع البيانات فأن مستند SMIL يمكنه وضع عناصر رسومية منفصلة على عرض كائنات الوسائط المتعددة والروابط الخاصة بها فمثلا في نفس وقت تشغيل الصوت والصورة فإن النص الخاص بالروايات يمكن أن يصاحب العرض.

تطبيق HTML+TIME

يعمل SMIL بشكل مستقل عن صفحة الويب. كما أن عناصر الوسائط المتعددة التي تظهر عبر SMIL تظهر على لوح منفرد في إطار المستعرض حيث لا تتداخل مع المحتويات الخاصة بالصفحة فمثلا يسمح لك SMIL بوضع الوقت على العناصر السمعية والمرئية والنص ولا تسمح لك بإضافة معلومات زمنية إلى علامات HTML الأساسية مثل <LI> <P> أو <IMG> ويقوم SMIL بعمل نسخ من بعض العناصر مثل كيفية تحديد وضع العناصر على الصفحة.

وقد قدمت مايكروسوفت مع كل من ماكروميديا وكومباك تطبيق ينافس XML بشكل جزئي اسمه Timed Interactive Multimedia Extensions for HTML أو باختصار وقد صمم HTML+TIME على SMIL لتدعيم تزامن عناصر HTML التقليدية فمثلا يسمح HTML+TIME بكتابة صفحة ويب تضيف إلى الصفحة كلما تقدم الوقت مثل تعليمات البرمجة ٢-٤.

تعليمات البرمجة ٢-٤: SEASON يحتوي على تابعي LEAGUE

```
<html>
  <head><title>Countdown</title></head>
  <body>
    <p t:begin="0" t:dur="1">10</p>
    <p t:begin="1" t:dur="1">9</p>
    <p t:begin="2" t:dur="1">8</p>
    <p t:begin="3" t:dur="1">7</p>
    <p t:begin="4" t:dur="1">6</p>
    <p t:begin="5" t:dur="1">5</p>
    <p t:begin="6" t:dur="1">4</p>
    <p t:begin="7" t:dur="1"3</p>
    <p t:begin="8" t:dur="1">2</p>
    <p t:begin="9" t:dur="1">1</p>
    <p t:begin="10" t:dur="1">Blast Off!</p>
  </body>
</html>
```

هذه النماذج في غاية الفائدة عند الشروع في عمل اختبارات صغيرة تعتمد على الوقت أو ما شابه ذلك. ويظهر المثال الخاص بمهرجان الفيلم السابق ذكره في تعليمات البرمجة ٢-٣ عند استخدام هذا البرنامج على هذا النحو:

```
<t:seq id="kubrick">
  <t:audio src="beethoven9.mid"/>
  <t:video src="corange.mov"/>
  <t:textstream src="clockwork.htm"/>
  <t:audio src="zarathustra.mid"/>
  <t:video src="2001.mov"/>
  <t:textstream src="aclarke.htm"/>
</seq>
```

وهو قريب الشبه من إصدار SMIL على الرغم من وجود بعض الاختلافات. ولكن الاختلاف الكبير هو أن SMIL يتم حفظه في ملفات مستقلة وتستخدم بواسطة برامج محددة مثل RealPlayer ولكن إصدار HTML+TIME يفترض إن يتم تضمنيه في صفحة الويب ويستخدم بواسطة المستعرض. اختلاف آخر هو أن هناك العديد من المنتجات التي بإمكانها تشغيل ملفات

SMIL بما في ذلك RealPlayer G2 إلا أنه لا يوجد في الوقت الحالي مستعرضات يمكنها تشغيل HTML+TIME ولكن نأمل أن تتضمن الإصدارات القادمة من Internet Explorer دعماً خاصاً لبرنامج HTML+TIME.

هنالك بعض المميزات في HTML+TIME إلا أن W3C قد قامت بإضفاء المزيد من المميزات على SMIL أكثر من التي قدمتها مايكروسوفت في منتجها HTML+TIME. لذا يبدو أنها معركة عالمية بين مايكروسوفت وكل من تسول له نفسه الدخول في هذا الميدان يحاول كل طرف من الأطراف تقديم ما يمكنه من أدوات لجعل عملية تطوير صفحات مواقع الإنترنت تتم بمزيد من السهولة واليسر.

تطبيق Open Software Description

يعد تنسيق Open Software Description من أحد التطبيقات التي تنتمي إلى عائلة XML وقد قام بتطويره كل من ماريمبا ومايكروسوفت لتحديث البرامج آليا. يقوم OSD بتحديد علامات XML التي تصف مكونات البرامج. ويتضمن وصف المكونات إصدار المكون وبنيته وعلاقته بالمكونات الأخرى ومقدار اعتماده عليها. مما يؤدي إلى تزويد OSD بالمعلومات الكافية لتحديد ما إذا كان المستخدم يحتاج إلى تحديث من عدمه، وإذا كانت هناك حاجة إلى إجراء مثل هذه العملية من التحديث، فإنه من الممكن أن يتم تنفيذها بشكل آلي بدلاً من طلب ذلك من المستخدم. توضح تعليمات البرمجة ٢-٥ مثالاً لملف OSD لتحديث WhizzyWriter 1000:

تعليمات البرمجة ٢-٥: ملف OSD لتحديث WhizzyWriter 1000

```
<?XML version="1.0"?>
<CHANNELHREF="http://updates.whizzy.com/updateChannel.html">
  <TITLE>WhizzyWriter 1000 Update Channel</TITLE>
  <USAGE VALUE="softwareUpdate"/>
  <SOFTPKGHREF="http://updates.whizzy.com/updateChannel.html"
        NAME="{46181F7D-1C38-22A1-3329-00415C6A4D54}"
        VERSION="5,2,3,1""
        STYLE="MSAppLogo5"
        PRECACHE="yes">
   <TITLE>WhizzyWriter 1000</TITLE>
   <ABSTRACT>
    Abstract: WhizzyWriter 1000: now with tint control!
```

```
    </ABSTRACT>
    <IMPLEMENTATION>
     <CODEBASE HREF="http://updates.whizzy.com/tinupdate.exe"/>
    </IMPLEMENTATION>
  </SOFTPKG>
</CHANNEL>
```

توجد المعلومات الخاصة بعملية التحديث في ملف OSD. يتم حفظ ملفات التحديث في أرشيف منفصل CAB أو ملفات تنفيذية مع تحميلها عند الحاجة. وقد دار الكثير من الجدل حول أهمية مثل هذه العملية حيث أن العديد من التحديثات الخاصة بالبرامج تسبب العديد من المشاكل بدلاً من حلها، مما جعل العديد من المستخدمين بتجنب تجربة مثل هذه التحديثات إلى أن تتم تجربتها على نطاق أوسع.

رسومات الاتجاهات القابلة لتغيير أبعادها

تتميز رسوم المتجهات عن الصور النقطية GIF وصور JPEG المستخدمة على الويب في الوقت الحالي للكثير من الصور بما في ذلك التخطيطات والرسوم المتحركة والصور المشابهة. مع ذلك فأن الكثير من رسومات الاتجاهات التقليدية مثل PDF وPostScript وEPS قد تم تصميمها على أساس الحبر على ورق بدلا من استخدام الوسائل الإلكترونية من الشاشة، وهذا هو السبب وراء أن PDF على الويب أقل قيمة من HTML على الرغم من أنه يحتوي على مجموعة أكبر من الرسوم البدائية. ويجب أن يدعم تنسيق رسومات المتجهات الخاصة بالويب الكثير من الميزات التي ليس لها معنى على الورق مثل الشفافية والنص المرتبط والعناصر المتحركة والألوان الإضافية لتمكين محركات البحث لاستخراج النص من الصور. ولا تكون هناك أي حاجة إلى استخدام أي من هذه العناصر عند التعامل مع PDF وPostScript.

قدم العديد من العملاء الكثير من عروض W3C من تطبيقات XML لرسومات المنتجات ويتضمن هذا:

- Precision Graphics Markup Language من IBM وAdobe وNetscape وSun.
- Vector Markup Language من مايكروسوفت وMacromedia وAutodesk وHewlett-Packard وVisio.
- Schematic Graphics من Central Laboratory من Research Councils.
- DrawML من Excosoft AB.

- Hyper Graphics Markup Language (HGML) من PRP و PCSL orange.

كل من هذه البرامج تعكس اهتمامات وخبرات محرريها فمثلا PGML يشبه PostScript ولكن ببناء جملة سمة عنصر XML مع استخدام Polish التي تخص PostScript. توضح تعليمات البرمجة ٦-٢ تضمين مثلث قرمزي في PGML.

تعليمات البرمجة ٢-٦: مثلث وردي في PGML

```
<?xml version="1.0"?>
<!DOCTYPE pgml SYSTEM "pgml.dtd">
<pgml>
  <group name="pinkTriangle" fillcolor="pink">
    <path>
      <moveto x="0" y="0"/>
      <lineto x="100" y="173"/>
      <lineto x="200" y="0"/>
      <closepath/>
    </path>
  </group>
</pgml>
```

وقد صممت W3C مجموعة عمل مع ممثلين من الوكلاء السابق ذكرهم لتحديد إحدى رسومات المتجهات الموحدة القابلة لتغيير الحجم المسمي SVG وهي تعد تطبيق من XML لوصف الرسومات الثنائية الأبعاد. وهي تعرف ثلاثة أنواع أساسية من الرسوم وهي الأشكال والصور والنصوص. ويتم تحديد النص عن طريق تحديد الحدود الخارجية الخاصة به كما يعرف أيضاً بالمسار الخاص به. والصور هي ملف صور نقطي مثل GIF أو JPEG. ويتم تعريف النص على أساس أنه سلسلة نصية لها خط معين كما أنه من الممكن أن يتم إلحاقه بالمسار، لذلك فإنها ليست محكومة بالخطوط الأفقية الخاصة بالنص التي تشبه النص العادي. كما أنه من الممكن أن يتم وضع كل هذه العناصر الثلاثة على الصفحة في مواقع محددة. ولأن SVG هو تنسيق خاص بالنصوص فإنه من السهل التعامل معه من أي برنامج. ويمكن على وجه الخصوص دمج كل من DHTML ECMAScript لعمل الصور على صفحات الويب التي تتفاعل مع ما يقوم به المستخدم من أفعال.

لأن SVG يصف رسومات أكثر من النص على العكس من أغلب تطبيقات XML المشروحة في هذا الفصل- فإنك في الأغلب ستحتاج إلى عرض برنامج خاص. وتفترض كل لغات ورقة

النمط المقدمة عرض بيانات نصية أساسية وكلها لا تدعم متطلبات الرسوم الضخمة من تطبيق مثل SVG. يمكن إضافة تدعيم SVG إلى أحد المستعرضات خاصة أن Mozilla يعد تعليمات برمجية مفتوحة وسيكون أسهل لكتابة المزيد من الإضافات إلا أنه في الوقت الحالي تعد الفائدة الأساسية من SVG هي أنه يستخدم لتبادل التنسيق بين البرامج المختلفة مثل CorelDraw وAdobe Illustrator والتي تستخدم تنسيقات ثنائية مختلفة.

لم يتم الانتهاء من SVG حتى كتابة هذه السطور. كما أنه لم يتم تنفيذ أياً منها وقد تم تحرير أول تجربة SVG بواسطة World Wide Web Consortium في فبراير ١٩٩٩. إلا أنه عند مقارنتها مع التجارب الأخرى فهي تعد غير متكاملة على الإطلاق فهي ليست سوى شكل عام لعناصر رسومية ستحتاج إلى تضمينها بدون أدنى قدر من التفاصيل حول كيفية كتابة هذه العناصر في XML.

لغة ترميز المنتجات "XML"

قامت مايكروسوفت بتطوير تطبيق XML الخاص بها لرسومات المنتجات المسمى لغة ترميز المتجهات والتي يرمز لها بالرمز VML وقد أوشك VML على الانتهاء بمقدار أكثر من SVG كما أنه قد تم تدعيمه بواسطة Internet Explorer 5.0 ومايكروسوفت ٢٠٠٠ . توضح تعليمات البرمجة ٢-٧ هي ملف HTML مع تضمين VML الذي يقوم برسم المثلث القرمزي. ويوضح الشكل ٢-٤ هذا الملف المعروض في Internet Explorer 5.0. ومع ذلك فأن VML ليس بنفس كانيات SVG ولا توجد به الكثير من الميزات المتقدمة الموجودة في SVG.

تعليمات البرمجة ٢-٧: المثلث القرمزي في VML

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">
  <head>
    <title>
      A Pink Triangle, Listing 2-7 from the XML Bible
    </title>
    <object id="VMLRender"
      classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
    </object>
    <style>
      vml\:* { behavior: url(#VMLRender) }
    </style>
```

```
 </head>
 <body>

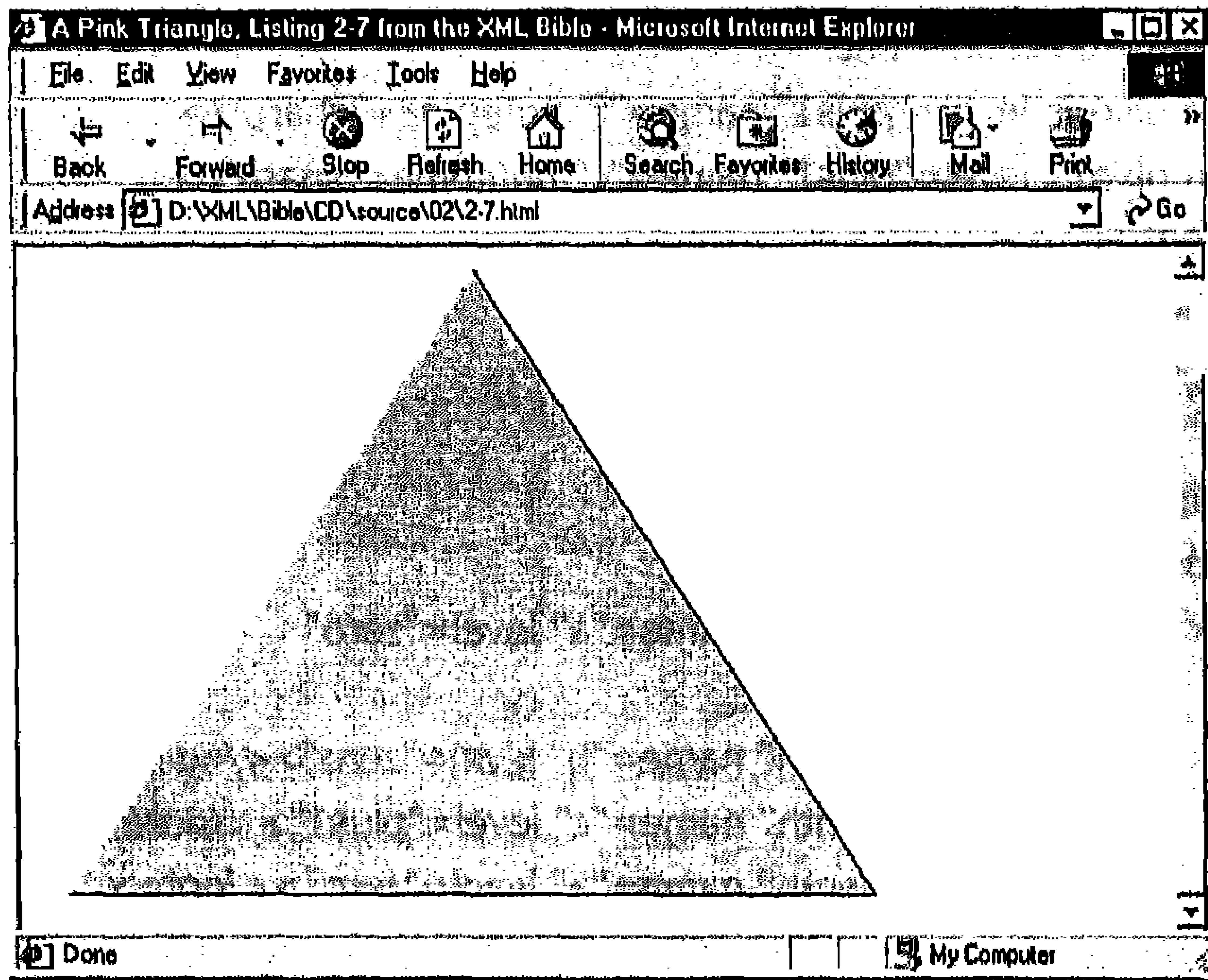

<div>

 <vml:polyline
  style="width: 200px; height: 200px"
  stroke="false"
  fill="true"
  fillcolor="#FFCCCC"
  points="10pt, 275pt, 310pt, 275pt, 160pt, 45pt">
 </vml:polyline>

</div>
</body>
</html>
```

ليس هناك ما يدعو لوجود إصداران قياسيان من رسومات المتجهات منفصلين على الويب. وفي أغلب الأحوال تقوم مايكروسوفت بتدعيم SVG في النهاية. وعلى الرغم من ذلك فإن VML متاح حالياً حتى إذا كان استخدامه محدد لمنتجات مايكروسوفت حيث أن SVG لم يكتمل بعد ويفضل رسامو الويب وجود قياس واحد على وجود اثنين.

وليس خارج البحث أن يتم تدعيم الاثنين بواسطة مستعرضي الويب على الأقل فإن XML المضمن يجعل ذلك أسهل للمبرمجين إن يكتبوا المحولين الذين يقوموا بترجمة الملفات من تنسيق إلى آخر. وطالما أن التنسيقات يتم توثيقها فإنه من الممكن أن تقوم مستعرضات الويب بدعم كل منهما. وعلى الأقل يقوم XML أسهل بالنسبة للمبرمجين لكتابة المحولات التي تحول الملف من أحد التنسيقات إلى تنسيق آخر.



الشكل ٢-٤ المثلث القرمزي اللون الذي تم إنشاءه بواسطة VML

المرجع يتم شرح VML بالتفصيل في الفصل ٢٢ "لغة الترميز المحورية".

MusicML

قام Connection Factory بإنشاء تطبيق XML لورقة الموسيقى المسماة MusicML وتتضمن MusicML الملاحظات والإيقاعات والصفوف والأصوات والسكنات والمقامات...الخ. توضح تعليمات البرمجة ٢-٨ الشريط الأول من مقطوعة Beth Anderson Flute Swale في MusicML.
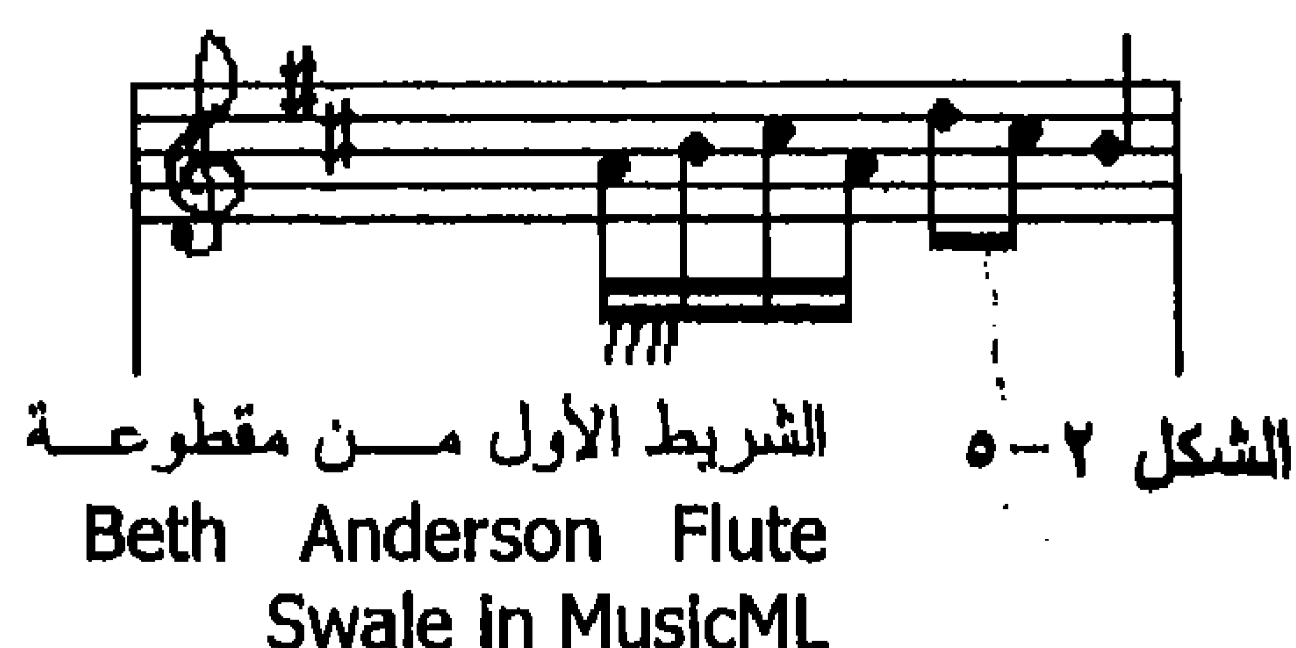
تعليمات البرمجة ٢-٨: شريط الأول من مقطوعة Beth Anderson

```
<?xml version="1.0"?>
<!DOCTYPE sheetmusic SYSTEM "music.dtd">
<sheetmusic>
 <musicrow size="one">
   <entrysegment>
```

```
  <entrypart cleff="bass" rythm="fourquarter"
          position="one">
   <molkruis level="plus1" name="f" notetype="sharp"/>
   <molkruis level="plus1" name="c" notetype="sharp"/>
  </entrypart>
 </entrysegment>
 &lt;segment&gt;

  <subsegment position="one">
   <beam size="double">
    <note beat="sixteenth" name="a" level="zero"
        dynamics="mf"/>
    <note beat="sixteenth" name="b" level="zero"></note>
    <note beat="sixteenth" name="c" level="plus1"></note>
    <note beat="sixteenth" name="a" level="zero"></note>
   </beam>
   <beam size="single">
    <note beat="eighth" name="d" level="plus1"/>
    <note beat="eighth" name="c" level="plus1"/>
   </beam>
   <note beat="quarter" name="b" level="zero"/>
   <note beat="quarter" name="a" level="zero"/>
  </subsegment>
 &lt;/segment&gt;
</musicrow>
</sheetmusic>
```

وقد قام Connection Factory بكتابة برنامج Java صغير يمكنه تحليل وعرض MusicML. يعرض الشكل ٢-٥ المثال السابق الذي تم عمله بواسطة هذا البرنامج. إلا أن به بعض العيوب "على سبيل المثال الملاحظة الأخيرة غير موجودة" غير أنها بوجه عام تعد من الأشياء الجيدة.



الشكل ٢-٥ الشريط الأول مـــن مقطوعـــة Beth Anderson Flute Swale in MusicML

لا يقوم MusicML بتبديل Finale أو Nightingale في أي وقت. وهي تبدو كأنـــها مفـــهوم وليس على هيئة منتج كما أنها تحتوي على العديد من المشاكل التي سوف تؤدي إلـــى اســـتثارة غضب الموسيقيين "فعلى سبيل المثال تجد أن الإيقاع قد تمت كتابته على نحو غير مضبوط ومـــا إلى ذلك".

يعد مثل هذا الأمر تنسيقاً خارجياً مقبولاً لبرامج تدوين الرموز الموسيقية الـــذي يمكـــن بـــه عرض ورقة الموسيقى على الويب. وبالإضافة إلى ذلك إذا كانت كل برامـــج تدويـــن الرمـــوز الموسيقية تدعم كلها MusicML فإنه من الممكن استخدامها كتنسيقات تبادلية لنقل البيانات من أحد البرامج إلى برنامج آخر وهو ما قد يحتاجه العديد من الموسيقيين في الوقت الحالي.

تطبيق VoxML

يعد Motorola VoxML (http://www.voxml.com) تطبيق XML للكلمة وقد تم تصميم هذا البرنامج على وجه الخصوص للرسائل التي ترد عبر البريد الصوتي أو نظم الرد على التليفون.

يقوم Voxml بتمكين نفس البيانات التي تستخدم على موقع الويب لتقدم عبر الهاتف وهـــي فـــي غاية الفائدة للمعلومات التي تم إنشائها عن طريق ربط جزأين من البيانات الصغيرة مثل أســـعار الأسهم ونتائج المباريات وتقرير حالات الطقس ونتائج الاختبارات. ويستخدم كل من Weather Channel CBS MarketWatch.com التطبيق VoxML للتزويد بالمزيد من المعلومات حـــول الاتصالات الصوتية التليفونية المعتادة.

وقد يبدو أحد الملفات الصغيرة للنظام التليفوني الآلي لإحدى الشركات مثل تعليمـــات البرمجـــة الموجودة في تعليمات ٢-٩ البرمجية.

تعليمات البرمجة ٢-٩: ملف VoxML

```
<?xml version="1.0"?>
<DIALOG>
  <CLASS NAME="help_top">
```

```
  <HELP>Welcome to TIC consumer products division.
      For shampoo information, say shampoo now.
  </HELP>
</CLASS>

<STEP NAME="init" PARENT="help_top">
  <PROMPT>Welcome to Wonder Shampoo
   <BREAK SIZE="large"/>
   Which color did Wonder Shampoo turn your hair?
   </PROMPT>
   <INPUT TYPE="OPTIONLIST">
    <OPTION NEXT="#green">green</OPTION>
    <OPTION NEXT="#purple">purple</OPTION>
    <OPTION NEXT="#bald">bald</OPTION>
    <OPTION NEXT="#bye">exit</OPTION>
  </INPUT>
</STEP>

<STEP NAME="green" PARENT="help_top">
  <PROMPT>
   If Wonder Shampoo turned your hair green and you wish
   to return it to its natural color, simply shampoo seven
   times with three parts soap, seven parts water, four
   parts kerosene, and two parts iguana bile.
  </PROMPT>
  <INPUT TYPE="NONE" NEXT="#bye"/>
</STEP>

<STEP NAME="purple" PARENT="help_top">
  <PROMPT>
   If Wonder Shampoo turned your hair purple and you wish
   to return it to its natural color, please walk
   widdershins around your local cemetery
    three times while chanting "Surrender Dorothy".
```

```
    </PROMPT>
    <INPUT TYPE="NONE" NEXT="#bye"/>
  </STEP>

  <STEP NAME="bald" PARENT="help_top">
    <PROMPT>
     If you went bald as a result of using Wonder Shampoo,
     please purchase and apply a three months supply
     of our Magic Hair Growth Formula(TM). Please do not
     consult an attorney as doing so would violate the
     license agreement printed on inside fold of the Wonder
     Shampoo box in 3 point type which you agreed to
     by opening the package.
    </PROMPT>
    <INPUT TYPE="NONE" NEXT="#bye"/>
  </STEP>

  <STEP NAME="bye" PARENT="help_top">
   <PROMPT>
    Thank you for visiting TIC Corp. Goodbye.
   </PROMPT>
   <INPUT TYPE="NONE" NEXT="#exit"/>
  </STEP>

</DIALOG>
```

لا يمكن عرض لقطة على الشاشة من هذا المثال لأنه الهدف منه لم يكن العرض على مستعرض ويب ولكن الهدف منه هو أن تستمع إليه عبر الهاتف.

فتح تبادل مالي

لا يمكن تغيير البرامج فجأة فالبيانات التي يعرف برنامج كيف يقرأها بها بعض القصور الذاتي فكلما زادت البيانات الموجودة في حق ملكية برنامج وتنسيق غير موثق كلما كان من الصعب تغيير البرامج. على سبيل المثال، قمت بتخزين بيانات تعاملاتي المالية خلال الخمس سنوات

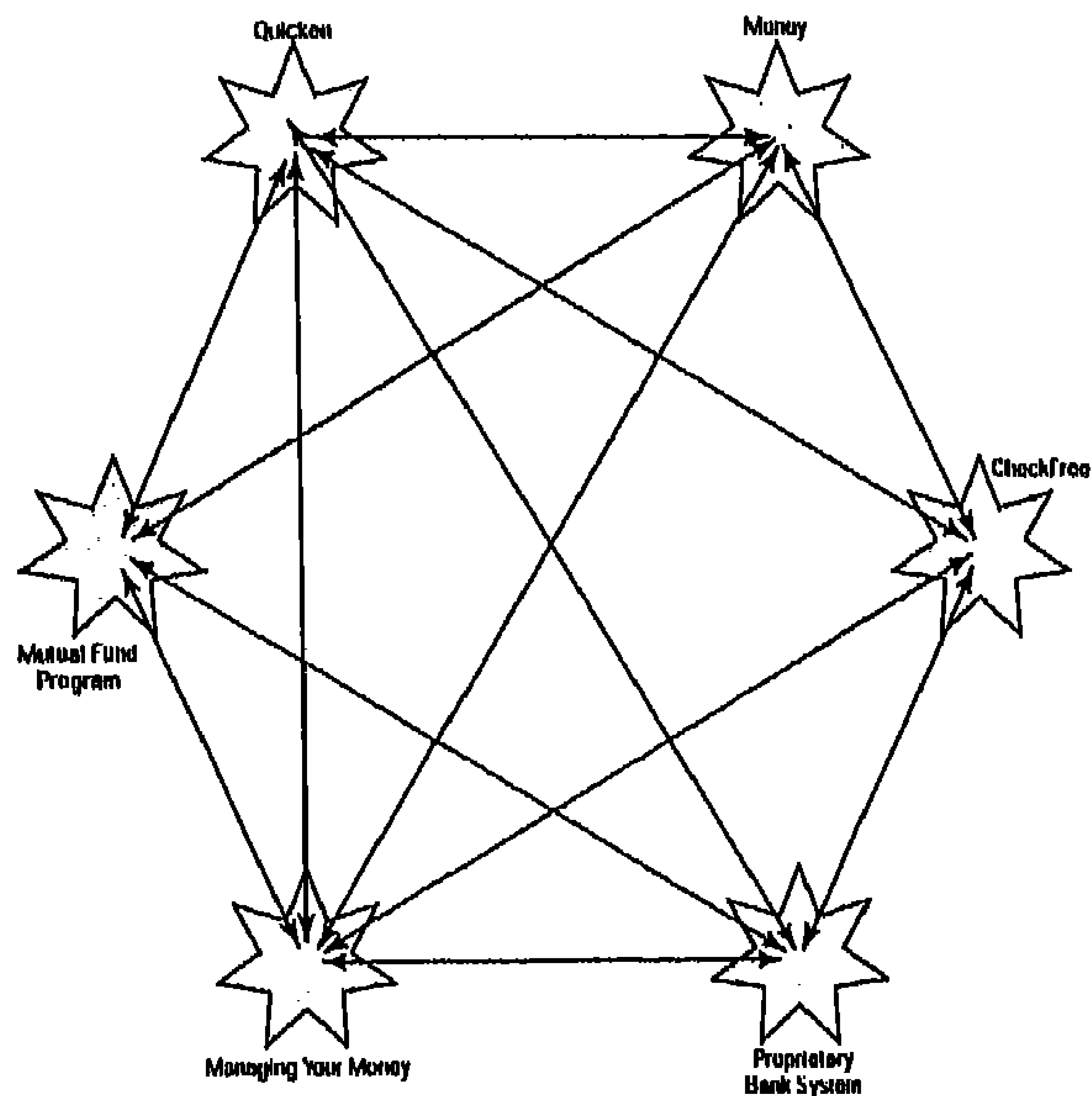
الماضية في Quicken، فهل من المحتمل أن أقوم بالتغيير إلى Microsoft Money حتى وإن كان لديها مميزات احتاج إليها لأنها لا توجد في Quicken ؟. لن يحدث ذلك إلا إذا كان برنامج Money يستطيع قراءة ملفات Quicken بدون أي خسارة في البيانات.

يمكن أن تحدث المشكلة داخل شركة واحدة أو منتجات شركة واحدة فعلى سبيل المثال لا يستطيع مايكروسوفت وورد ٩٧ الذي يخص ويندوز قراءة المستندات التي تم إنشاءها بواسطة إصدارات سابقة من ورد وكذلك الإصدارات السابقة من ورد لا يمكن أن تقرأ ملفات ورد ٩٧ على الإطلاق. أما بالنسبة لمايكروسوفت ٩٨ الذي يخصMac فهو لا يستطيع قراءة كل شيء في ورد٩٧ لملف ويندوز على الرغم من أن ورد ٩٨ الذي يخصMac تم إصدارة بعد عام واحد من إصدار ورد ٩٨.

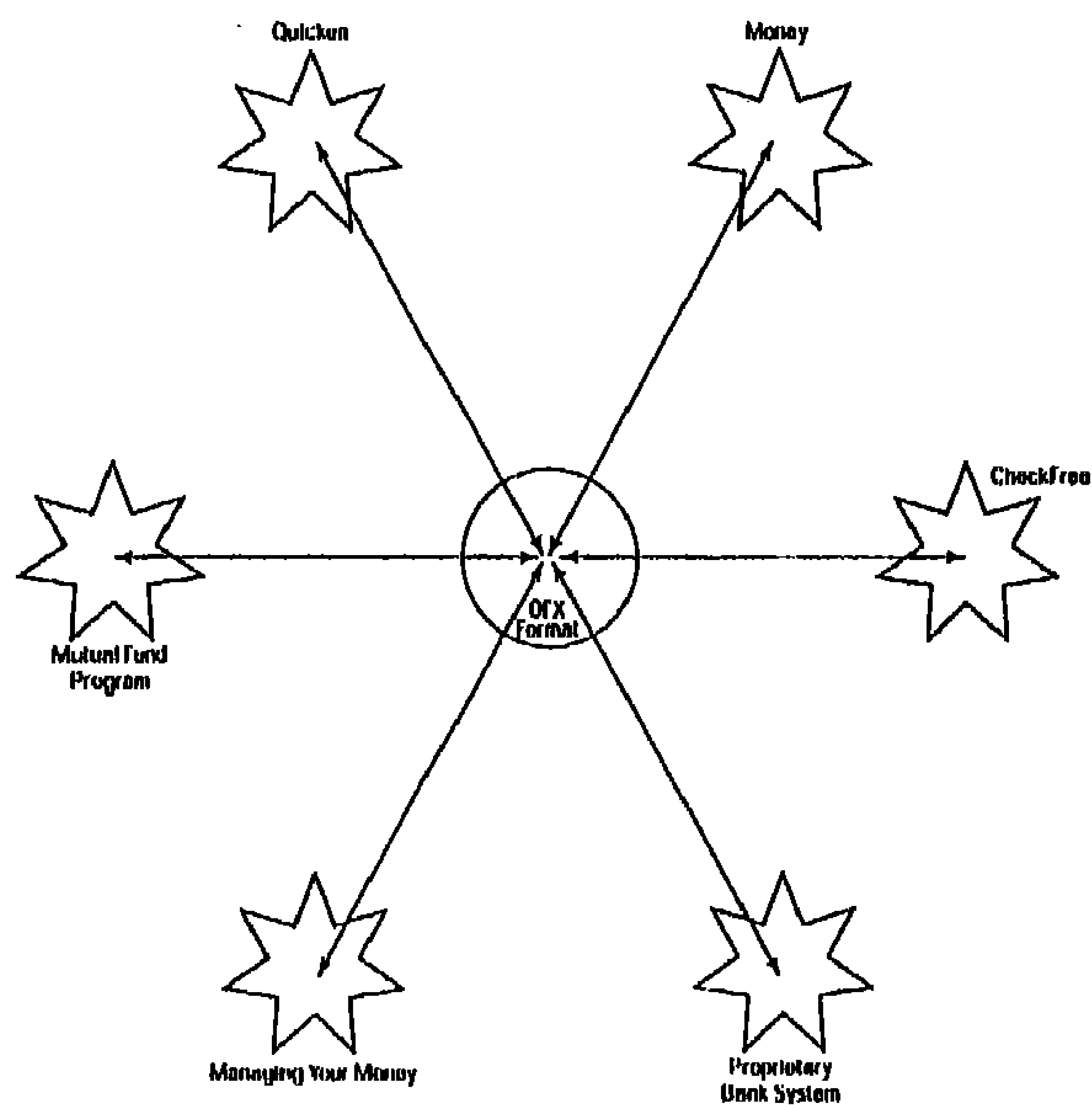
كما ذكر في الفصل الأول فان Open Financial Exchange Format (OFX) هو تطبيق XML يستخدم لوصف بيانات مالية من النوع الممكن تخزينه في منتج مالي شخصي مثل Money أو Quicken. يستطيع أي برنامج يفهم OFX قراءة بيانات OFX ولأن OFX قد تم توثيقه بالكامل وليس ذو ملكية خاصة "على عكس التنسيقات الثنائية لبرامج Money و quicken وغيرهم" فمن السهل على المبرمجين كتابة التعليمات البرمجية لفهم OFX.

لا يسمح OFX فقط بتبادل البيانات بين برنامج Money وبين برنامج Quicken بل يسمح لبرامج أخرى تستخدم نفس التنسيق بتبادل البيانات. على سبيل المثال، إذا أراد بنك أن يرسل كشوف حسابات للعملاء آليا يجب فقط أن يضع برنامج واحد فقد لتشفير كشوف الحسابات في تنسيق OFX بدلا من وضع عدة برامج لتشفير كشوف الحسابات في تنسيق Quicken وتنسيق Money وتنسيق Managing Your Money وغيرهم.

كلما زادت البرامج التي تستخدم تنسيقات سابقة الإعداد كلما انخفض المجهود والتكاليف المنفقة في التطوير.على سبيل المثال يتطلب من ستة برامج لتستطيع أن تقرأ وتكتب تنسيقات الملكية الخاصة بهما وبالبرامج الأخرى ٣٦ محول مختلف. ولكن تتطلب ستة برامج تقرأ وتكتب نفس تنسيق OFX ستة محولات فقط. وبذلك يتم تقليل المجهود من O(n) إلى O(n2). يصف الشكل ٢-٧ ستة برامج تقرأ وتكتب نفس تنسيق OFX. ويمثل كل سهم محول عليه تبادل الملفات والبيانات بين البرامج. يمكن أن ترى في الشكل ٢-٦ الروابط لستة برامج مختلفة تكتب وتفسر حق ملكية التنسيق الثنائي لكل منهما وفي الشكل ٢-٧ يمكن أن ترى نفس البرامج الستة المختلفة تقرأ وتكتب تنسيق XML مفتوح.يعتبر التبادل الذي يعتمد على XML أكثر سهولة من ... التنسيق الثنائي.



الشكل ٢–٦ ستة برامج مختلفة تكتب وتقرأ تنسيقها وتنسيق بعضها البعض



الشكل ٢–٧ ستة برامج تقرأ وتكتب نفس تنسيق OFX

لغة وصف النماذج الممتدة

ذهبت اليوم إلى المكتبة واشتريت نسخة من رواية الكاتب Armistead Maupin وهي Sure of You ودفعت ثمن هذه الرواية ببطاقة الائتمان وعندما فعلت ذلك قمت بتوقيع ورقة تثبت أنني سأدفع لشركة البطاقة الائتمانية مبلغ وقدرة ١٤,٠٧ دولار عندما تطلب مني ذلك وإذا رفضت دفع قيمة هذه البطاقة من حق الشركة إن تلجأ للقضاء مستخدمة هذه الورقة التي وقعت عليها لتثبت للمحكمة أنني قد وافقت على دفع المبلغ وهو ١٤,٠٧ دولار.

في نفس اليوم قمت بطلب رواية The Vampire Armand للكاتبة Anne Rice من المكتبة الفورية amazon.com. وقد بلغت التكلفة ١٦,١٧ دولار بالإضافة إلى مصاريف الشحن التي تبلغ ٣,٩٥ وأيضاً قمت بالدفع بالبطاقة الائتمانية ولكن الاختلاف أن Amazon لم تحصل مني على إمضاء على ورقة ومع ذلك سترسل لي شركة بطاقة الائتمان فاتورة بإجمالي حساب ما قمت بشرائه وسأقوم بدفعها ولكن إذا رفضت دفع الفاتورة فلا توجد لديهم ورق بإمضائي توضح أنني وافقت على دفع ٢٠ دولار يوم ٩٨/١٠/١٥ وإذا زعمت أنني لم أشتر شئ فإن شركة بطاقة الائتمان سترسل الحساب مرة ثانية إلى Amazon. لذلك قبل أن تقبل Amazon أو يقبل البائعين عبر الهاتف أو بالطرق الفورية الأخرى أن تتعامل ببطاقات الائتمان بدون الحصول على إمضاءات على ورق كما كان يحدث، يجب أن يقبلوا تحمل مسئولية النزاعات التي قد تحدث نتيجة مثل هذه المعاملات التجارية.

من الصعب وضع أرقام دقيقة، حيث توجد اختلافات من بائع إلى أخر، ولكن أقل من ١٠% من المعاملات المالية عبر الإنترنت يتم إرجاعها إلى البائع الأصلي بسبب الغش الذي يحدث في البطاقات الائتمانية والنزاعات التي تحدث بسببها. يعتبر هذا مقدار ضخم ولكن موقع مثل موقع Amazon يتقبل مثل ذلك كتكلفة للعمل على شبكة الإنترنت ويقومون بإدخال هذا الفاقد في هيكل أسعار المنتجات ولكن لن يصلح ذلك مع المعاملات المالية الكبيرة التي تتضمن ستة أرقام. فلن يرغب أي من المتعاملين بهذا الأسلوب في بيع مواد تقدر بحوالي ٢٠٠,٠٠٠ دولار ليدعي المشتري بعد ذلك أنه لم يقم بالشراء أو استلام الطلب. قبل ظهور المعاملات المالية عبر الإنترنت لابد من تطوير طريقة للتأكد من أن طلب الشراء قد تم تقديمه من شخص محدد وأن هذا الشخص هو بالفعل الشخص الصحيح، وليس ذلك فقط بل يجب أن تكون طريقة التأكد معترف بها في المحاكم.

جزء من حل المشكلة هو الإمضاءات الرقمية وهي المماثل الإلكتروني للحبر على الورق أي الإمضاء على الورق. للإمضاء رقميا على مستند، يمكن حساب التعليمات البرمجية hash للمستند باستخدام نوع حساب معروف وتشفير التعليمات البرمجية hash باستخدام المفتاح الخاص بك، وإرفاق التعليمات البرمجية hash المشفرة بالمستند. يمكن للمتعاملين فك شفرة تعليمات برمجة hash باستخدام المفتاح العام الخاص بك والتأكد من أنها توافق المستند. ومع

ذلك، لا يمكنهم توقيع المستندات بالنيابة عنك لأنهم لا يملكون المفتاح الخاص بك، والبروتوكول الدقيق المتبع هو أكثر تعقيداً في الممارسة ولكن الفكرة الأساسية هي أن مفتاحك الخاص مدمج مع البيانات التي تقوم بإمضائها بطريقة تسمح بالتأكد من صحته ولا يستطيع أحد ممن لا يعرفون مفتاحك الخاص إمضاء المستند بالنيابة عنك.

لا يعتبر هذا النظام محمي تماما فمن الممكن سرقة مفتاحك الخاصة على سبيل المثال ولكن من الصعب تزوير الإمضاء الرقمي بنفس درجة صعوبة تزوير الإمضاء الحقيقي على الورق. مع ذلك، هناك عدد محدود من الهجوم على بروتوكولات الإمضاء الرقمي وأحد أهم نقاط الهجوم هو تغيير البيانات التي تم إمضاءها، ويقوم تغيير البيانات التي تم إمضائها بإلغاء صحة الإمضاء ولكن لا يحدث ذلك إذا كانت البيانات التي تم تغييرها لم يتم تضمينها في المقام الأول. على سبيل المثال، عند تقديم نموذج HTML فالأشياء الوحيدة التي يتم إرسالها هي القيم التي تقوم بملئها في حقول النموذج وأسماء الحقول، أما بالنسبة لبقية ترميز HTML فلا يتم تضمينه. قد توافق على دفع ١٥٠٠ دولار لشراء جهاز كمبيوتر Pentium II 450 MHz جديد يقوم بتشغيل Windows NT ولكن الشيء الوحيد الذي يرسل على النموذج هو ١٥٠٠ دولار والإمضاء على هذا الرقم يعني أنك وافقت على دفع المبلغ ولكنك لم توضح ما نوع المشتريات ويستطيع البائع إرسال أي شيء غير البضاعة المتفق عليها ويزعم أنه ما تم الاتفاق عليه ولهذا إذا كان الإمضاء الرقمي سيكون مفيداً يجب توضيح تفاصيل المعاملات التجارية بالكامل فيه ولا يتم حذف شيء.

تسوء المشكلة إذا كنت ستتعامل مع الحكومة الفيدرالية الأمريكية، فالقواعد الحكومية في طلبات الشراء والتوريد تستلزم كتابة محتويات النماذج بالتفصيل الدقيق بدء من أدق وأصغر التفاصيل وصولا إلى واجهة الخط وحجم النوع، وأي فشل في سرد المواصفات الخاصة بالمنتج في دقة متناهية قد يعرض مثلا شحنة أسلحة يورانيوم تقدر قيمتها بحوالي ٢٠ مليون دولار أن يتم رفضها بسبب عدم دقة سرد المواصفات. لذلك تحتاج ليس فقط إلى توضيح ما تم الاتفاق عليه بدقة ولكن تحتاج أيضا إلى توضيح أنك قد نفذت كافة المتطلبات القانونية للنموذج. ونماذج HTML ليست بالتطور الكافي لمعالجة مثل هذه الاحتياجات.

ومع ذلك بإمكان XML في أغلب الأحيان أن يطور لغة الترميز لتصبح من القوة والتأثير بحيث تفي باحتياجات المستخدم وهذا المثال لا يعد استثناء. قدمت UWI.COM تطبيق XML يسمى Extensible Forms Description Language (XFDL) للنماذج التي لها متطلبات قانونية بالغة الخطورة والتي يجب إمضاءها بإمضاءات رقمية. يقدم XFDL الخيار لإجراء بعض العمليات الحسابية البسيطة في النموذج، على سبيل المثال تستطيع تعبئة ضريبة المبيعات والشحن ومصاريف المعاملات ثم بعد ذلك يتم تجميع الإجمالي.

UWI.COM قدم XFDL إلى W3C ولكنه غير مناسب لمستعرض ويب ولذلك لن يتم تطبيقه فيها. تكون الفائدة الحقيقية وراء استخدام XFDL إذا تم تطبيقها في المعاملات التجارية buisness-to-buisness و buisness-to-government. بإمكان XFDL أن تصبح جزء أساسي من التجارة الإلكترونية فيما بعد.

لغة ترميز الموارد البشرية "HRML"

تعتبر لغة Human Resources Markup Language (HRML) الخاصة بشركة HireScape تطبيق XML الذي يقدم معجم بسيط لوصف أساسيات وظيفة فهو يصف العناصر التي توافق أجزاء من إعلانات التي تطلب أفراد للعمل في الشركات والأقسام الداخلية وشروط والخبرة المطلوبة وغير ذلك. قد تبدو تعليمات.برمجة عمل الوظيفة في HRML مثل التعليمات البرمجية في تعليمات برمجة ٢-١٠.

تعليمات البرمجة ٢-١٠: تعليمات برمجة وظيفة ل HRML

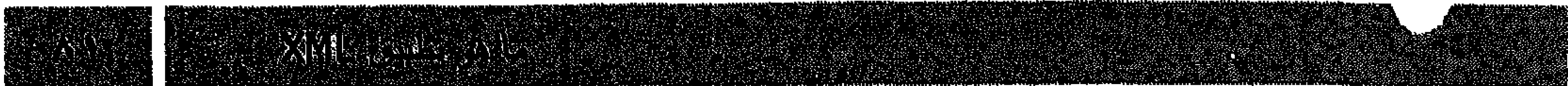
```
<?xml version="1.0"?>
<HRML_JOB>

  <COMPANY>

    <CO_NAME>IDG Books</CO_NAME>
    <CO_INTERNET_ADDR>
      <CO_HOME_PAGE>http://www.idgbooks.com/</CO_HOME_PAGE>
      <CO_JOBS_PAGE>
        http://www.idgbooks.com/cgi-
bin/gatekeeper.pl?uidg4841:%2Fcompany%2Fjobs%2Findex.html
      </CO_JOBS_PAGE>
    </CO_INTERNET_ADDR>
</COMPANY>

  <JOB>

    <JOB_METADATA>
```



```
<JOB_LOADED_DT>09/10/1998</JOB_LOADED_DT>
<JOB_LOADED_URL>
 http://www.idgbooks.com/cgi-
bin/gatekeeper.pl?uidg4841:%2Fcompany%2Fjobs%2Findex.html
</JOB_LOADED_URL>
</JOB_METADATA>

<JOB_DATA>

<JOB_TITLE>Web Development Manager</JOB_TITLE>

<JOB_NUMBER_AVAIL>1</JOB_NUMBER_AVAIL>
<JOB_YEARS_EXP>3</JOB_YEARS_EXP>
<JOB_DESC>
 This position is responsible for the technical
 and production functions of the Online
 group as well as strategizing and implementing
 technology to improve the IDG Books web sites.
 Skills must include Perl, C/C++, HTML, SQL, JavaScript,
 Windows NT 4, mod-perl, CGI, TCP/IP, Netscape servers
 and Apache server. You must also have excellent
 communication skills, project management, the ability
 to communicate technical solutions to non-technical
 people and management experience.
</JOB_DESC>

<JOB_KEYWORDS>
 Perl, C/C++, HTML, SQL, JavaScript, Windows NT 4,
 mod-perl, CGI, TCP/IP, Netscape server, Apache server
</JOB_KEYWORDS>

<JOB_TERMS PAY="Salaried" TYPE="Full-time">
 $60,000
</JOB_TERMS>
```

```
<JOB_LOCATION CITY="Foster City" STATE="California"
 STATE_ABBR="CA" POSTAL_CODE="94404" COUNTRY="USA">
</JOB_LOCATION>

</JOB_DATA>

</JOB>

<RESPONSE>

<RESP_EMAIL>cajobs@idgbooks.com</RESP_EMAIL>
<POSTAL_ADDR ENTITY_TYPE="response">
Continued
Listing 2-10ت(continued)
<ADDR_LINE_1>Dee Harris, HR Manager</ADDR_LINE_1>
<ADDR_LINE_2>919 E. Hillsdale Blvd.</ADDR_LINE_2>
<ADDR_LINE_3>Suite 400</ADDR_LINE_3>
<CITY>Foster City</CITY>
<STATE>CA</STATE>
<POSTAL_CODE>94404</POSTAL_CODE>
</POSTAL_ADDR>

</RESPONSE>

</HRML_JOB>
```

بالرغم من إمكانية تحديد ورقة نمط للغة HRML واستخدامها في وضع قوائم الوظائف المطلوبة على صفحات ويب إلا أن هذا ليس هو الهدف الأساسي منها. إنما تم تصميم HRML لجعل تبادل معلومات التوظيف بين الشركات والراغبين في العمل وغيرهم آليا. يوجد الآن على الإنترنت مئات الوظائف بالإضافة إلى قوائم البريد ومجموعات الأخبار التي تستخدم الإنترنت Usenet ولذلك من الصعب على فرد بمفرده أن يبحث في ذلك كله وكذلك من الصعب على جهاز الكمبيوتر أن يبحث عنهم جميعاً لأنهم جميعا يستخدموا تنسيقات مختلفة للرواتب والمواقع وكل ما شابه ذلك.

أما في حالة استخدام مواقع عديدة للغة HRML سيصبح من السهل نسبياً للباحث عن وظيفة أن يبحث باستخدام معيار مثل "كل الوظائف لمبرمجي Java في مدينة نيويورك التي تدفع أكثر من ١٠٠,٠٠٠ دولار سنويا مع وجود الضمان الصحي، وبذلك يمكن أن تدخل IRS بحث لوظيفة لكامل الوقت أو مترجم بالقطعة أو غير ذلك .

في الواقع العملي، يتم تنفيذ مثل هذه البحوث من خلال نموذج HTML مثلما يحدث عند البحث في شبكة الويب حاليا، والفرق الرئيس بين الطريقتين هو أن مثل هذا البحث قد يرجع نتائج أكثر إفادة لإمكانية استخدام هيكل البيانات وبناء جمل الترميز بدلا من الاعتماد على نص إنجليزي غير دقيق.

Resource Description Framework (RDF)

يضيف XML بنية إلى المستندات، ويعتبر Resource Description Framework (RDF) تطبيق XML الذي يضيف دلالات الألفاظ "معني Semantics". يمكن استخدام RDF لتحديد أي شئ بدء من الكاتب واستخراج أجزاء من صفحة الويب إلى الإصدار وحزمة البرامج المدمجة معه إلى مخرج وكاتب الشاشة والممثلين في فيلم سينمائي. الارتباطات التي يستخدمها هؤلاء هي الارتباطات المشفرة في RDF وليست البيانات نفسها "صفحة الويب والبرنامج والفيلم" ولكن معلومات بشأن البيانات. تسمي هذه البيانات meta-data "بيانات تفصيلية" وهي سبب وجود RDF.

يعرف معجم RDF مجموعة من العناصر والمحتوى المسموح به المناسب للبيانات التفصيلية في مجال معروف. يتيح RDF للمجموعات ذات الاهتمام الواحد جعل معاجمهم "مجموعة المصطلحات التي يستخدموها" قياسية ومشاركة هذه المعاجم مع آخرين يمكنهم توسيعها وزيادة حجمها. على سبيل المثال، عبارة Dublin Core هي معجم RDF صمم خصيصا للبيانات التفصيلية عن صفحات الويب، ويبني نظام Educom's Instructional Metadata System (IMS) على Dublin Core بإضافة عناصر تكون مفيدة عند وصف محتوى متصل بالمدرسة مثل مستوى التعليم وأهداف التعليم والتكلفة.

بالطبع، على الرغم من إمكانية استخدام RDF لأنظمة الطبع والنشر وكتالوجات تخزين صور الفيديو وتحديث البرامج آليا وكل ما شابه ذلك فأنه من الأفضل أن يتم استخدامه أولا في تضمين البيانات التفصيلية في صفحات الويب. ولدي RDF القدرة على تزامن علامات <META> الحالية المستخدمة في خرائط الموقع ومعدل المحتوى والفهرسة الآلية والمكتبات الرقمية في مجموعات موحدة تفهمها كل تلك الأدوات. بمجرد أن تصبح البيانات التفصيلية RDF جزء قياسي من صفحات الويب فستتمكن محركات البحث من إرجاع نتائج أكثر تركيزاً وأكثر إفادة. بذلك يستطيع العملاء الأذكياء الدخول إلى الويب لإيجاد المعلومات المطلوبة أو إجراء أعمال بالنيابة

عنك وهكذا تتحول الويب من وضعها الحالي كبحر غير مرتب من المعلومـــات إلـــى مخـــزون بيانات له بنية ويمكن البحث فيه وفهمه.

كما يوضح الاسم فإن RDF تصف مصادر. والمصدر هو أي شئ يكون لـــه URL. يتكـــون وصف مصادر من عدد من الخصائص ولكل خاصية نوع وقيمة، على سبيل المثـــال، <DC:Format>HTML</DC:Format> لها نوع "DC:Format" والقيمة "HTML". قد تكون هـذه القيم سلاسل نص أو أرقام أو تواريخ وغيرهم من المصادر. يمكن أن يكـــون لـــهذه المصـــادر الأخرى وصفهم الخاص في RDF، على سبيل المثال، تستخدم التعليمات البرمجية في تعليمـــات برمجة ٢-١١ معجم Dublin Core لوصف موقع Café Con Leche.

تعليمات البرمجة ٢-١١: وصف RDF الصفحة الرئيسية Con Leche باستخدام معجم Dublin Core

```
<RDF:RDF
  xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:DC="http://purl.org/DC/">

  <RDF:Description about="http://metalab.unc.edu/xml/">
    <DC:Creator>Elliotte Rusty Harold</DC:Creator>
    <DC:Language>en</DC:Language>
    <DC:Format>HTML</DC:Format>
    <DC:Date>1999-08-19</DC:date>
    <DC:Type>home page</DC:Type>
    <DC:Title>Cafe con Leche</DC:Title>
  </RDF:Description>
</RDF:RDF>
```

يتم استخدام RDF لإصدار ٢,٠ مـــن Platform for Internet Content Selection (PICS) وP3P) Platform for Privacy Preferences) وأيضا العديد من المناطق الأخـــرى التي تتطلب بيانات تفصيلية لوصف صفحات ويب ومحتويات أخرى.

XML for XML

يعتبر XML تنسيق بأهداف شديدة العمومية لبيانات النص وبعض الأشياء التي يستخدم من أجلـــها هي إضافة المزيد من التطوير على XML نفسه وهذا يتضمن لغة ورقة نمط XSL ولغـــة ربـــط XLL ووصف Document Content Description للغة XML.

لغة XSL

تعتبر لغة Extensible Style Language (XSL) تطبيق XML. لدى XSL جزأين أساسـيين، يعرف الجزء الأول معجم لتحويل مستندات XML ويتضمن هذا الجزء من XSL علامـات XML للأشجار والعقد والنقوش والقوالب وعناصر أخرى مطلوبة لمطابقة وتحويل مستندات XML من معجم ترميز إلى آخر "أو حتى إلى نفس الترميز ولكن بترتيب مختلف".

يعرف الجزء الثاني من XSL معجم XML لتنسيق مستند XML الذي تـم تحويلـه والمنتـج بواسطة الجزء الأول وهذا يتضمن علامات XML لتنسيق الكائنات بما في ذلك الكتل والأحـرف والقوائم والرسومات وغير ذلك. تعرض تعليمات برمجية ٢-١٢ ورقة نمط XSL مثالية.

تعليمات البرمجة ٢-١٢: ورقة نمط XSL

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:fo="http://www.w3.org/TR/WD-xsl/FO"
  result-ns="fo">
  <xsl:template match="/">
    <fo:basic-page-sequence >
      <xsl:apply-templates/>
    </fo:basic-page-sequence>
  </xsl:template>

  <xsl:template match="ATOM">
    <fo:block font-size="10pt" font-family="serif"
     space-before="12pt">
      <xsl:value-of select="NAME"/>
    </fo:block>
  </xsl:template>

</xsl:styleshee
```

المرجع نتعرف على المزيد عن XSL في الفصل ١٤ و ١٥.

لغة XLL

تعرف لغة (Extensible Linking Language (XLL رابطة جديدة أكثر عمومية تسمي Xlink. تقوم Xlink بإنجاز كل ما هو متاح مع الارتباطات التشعبية والارتساء الذي يعتمد على URL للغة HTML. مع ذلك، يمكن لأي عنصر أن يصبح ارتباط وليس فقط عنصر A. على سبيل المثال، عنصر footnote يمكن ربطه مباشرة بنص ملحوظة مثل ذلك:

```
<footnote xlink:form="simple" href="footnote7.xml"٧<</footnote>
```

بالإضافة إلى ذلك، تستطيع Xlink أن تفعل الكثير من الأشياء التي لا تستطيع ارتباطات HTML أن تفعلها. فبإمكان Xlinks استخدام اللغة العربية ليستطيع القراء الرجوع إلى الصفحة التي بدءوا منها. بإمكان Xlink الارتباط بمواقع عشوائية في المستند ويمكن في Xlinks تضمين نص أو بيانات رسومية داخل مستند بدلا من أن تطلب من المستخدم أن يقوم بتشغيل الرابطة "وهي شيء مثل علامة <IMG> في HTML ولكنها أكثر مرونة". باختصار، بإمكان Xlink أن تجعل النص المترابط أكثر فاعلية.

المرجع يتناول الفصل ١٦ "Xlinks" بالتفصيل.

DCD

تتراوح التسهيلات التي تقدمها لغة XML لتعريف كيف يتم تنسيق محتويات عنصر XML من ضعيفة إلى غير موجودة. على سبيل المثال، افترض كجزء من تاريخ أنك قمت بإعداد عناصر MONTH هكذا:

```
<MONTH>9</MONTH>
```

كل ما يمكن قوله هو أن محتوى عنصر MONTH يجب أن يكون بيانات حرف، ولا يمكن القول أن الشهر يجب أن يكون عدد صحيح بين ١ و١٢.

تم اقتراح العديد من الأنظمة لاستخدام XML نفسه لتحديد ما الذي يظهر في محتويات أي عنصر. أحد هذه الاقتراحات هو (Document Content Description (DCD. على سبيل المثال، هناك DCD يخبرك أن عناصر MONTH تحتوي فقط على عدد صحيح بين ١ و ١٢.

```
<DCD>
  <ElementDef Type="MONTH" Model="Data" Datatype="i1"
   Min="1" Max="12"/>
</DCD>
```

هناك العديد من الأمثلة الممكن عرضها لاستخدام XML من أجل XML ولكن الأمثلة السابق عرضها توضح النقطة الأساسية وهي أن لغة XML كافية لوصف وشرح نفسها. ومن ضمن أشياء أخرى، هذا يعني أن مواصفات XML يمكن أن تظل بسيطة وصغيرة. قد لا يكون هناك إصدار XML 2.0 لأن أي إضافات أساسية أخرى مطلوبة يمكن بناءها من XML الخام بدلا من جعلها المميزات الجديدة في XML. يمكن للأشخاص والبرامج التي تحتاج هذه المميزات المطورة الجديدة استخدامها، والأخريين الذين لا يحتاجونها بإمكانهم تجاهلها ولا تحتاج لمعرفة ما لا تستخدمه وتقدم لك لغة XML الطرق التي يمكنك بها بناء أي شئ.

الاستخدامات الخفية XML

ليست كل تطبيقات XML قياسات مفتوحة وشائعة الاستخدام. يتحرك الكثير من بائعي البرامج باتجاه XML لبياناتهم لأنها تنسيق سهل الفهم وعام للبيانات الهيكلية التي يمكن تشغيلها والتعامل معها بأدوات رخيصة ومجانية.

يقوم مايكروسوفت أوفيس ٢٠٠٠ بتطوير HTML إلى تنسيق ملف coequal مع تنسيقات ثنائية الأصل. مع ذلك، لا يقوم HTML 4.0 بتدعيم كل المميزات التي يطلبها برنامج أوفيس مثل تتبع المراجعات، التعليقات، الفهرس، وغير ذلك. البيانات الإضافية التي لا يمكن كتابتها على أساس أنها لغة HTML يتم تضمينها في ملف في أجزاء صغيرة من XML. رسومات ورد vector graphics المخزنة في VML. في هذه الحالة، يعتبر العامل الأساسي هو عدم رؤية XML المضمن XML's invisibility في المستعرض القياسي. أشياء هامة.

تستخدم Federal Express معلومات تتبع تفصيلية كميزة تنافسية إضافية على شركات الشحن الأخرى مثل UPS وPost Office. أولا تتاح هذه المعلومات من خلال برنامج مخصص ثم من خلال الويب. بدأت FedEx حاليا اختبار بيتا API/library التي بإمكان طرف ثالث ومطورين داخليين استخدامهم لتوحيد برامجهم وأنظمتهم مع FedEx's والتنسيق المستخدم لهذه الخدمة هو XML.

يدعم Netscape Navigator 5.0 العرض المباشر XML في مستعرض الويب، ولكن بدأ Netscape بالفعل باستخدام XML داخلياً كما حدث في إصدار ٤,٥. عندما تطلب من Netscape أن تعرض قائمة للمواقع المرتبطة بالموقع الحالي الذي تبحث فيه، فان المستعرض يصلك برنامج CGI الذي يعمل على خادم Netscape وتكون البيانات التي يرسلها الخادم إليك بلغة XML. توضح تعليمات برمجة ٢-١٣ بيانات XML للمواقع المرتبطة بالموقع .http://metalab.unc.edu/

تعليمات البرمجة ٢-١٢: بيانات XML للمواقع المرتبطة بموقع http://metalab.unc.edu/

```
<?xml version="1.0"?>
<RDF:RDF>
<RelatedLinks>
<aboutPage
href="http://info.netscape.com/fwd/rl/http://metalab.unc.edu:80/*">
</aboutPage>
<child instanceOf="Separator1"></child>
<child
href="http://info.netscape.com/fwd/rl/http://www.sun.com/"
name="Sun Microsystems">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://www.unc.edu/"
name="unc">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://sunsite.sut.ac.jp/"
name="SunSITE Japan">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://sunsite.nus.sg/"
name="SunSITE Singapore">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://sunsite.berkeley.edu/"nam
e="Berkeley Digital Library SunSITE">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://www.sun.com/sunsite"nam
e="SunSITE on the net">
</child>
<child
```

```
href=http://info.netscape.com/fwd/rl/http://www.sunsite.auc.dk/"
name="SunSITE Denmark">
</child>
<child href="http://info.netscape.com/fwd/rl/http://sunsite.edu.cn/"
name="SunSITE China">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://sunsite.stanford.org/"
name="Stanford University SunSITE">
</child>
<child
href="http://info.netscape.com/fwd/rl/http://www.cdromshop.com/cdsh
op/desc/p.061590000085.html" name="SunSITE Archive">
</child>
<child instanceOf="Separator1"></child>
<child instanceOf="Separator1"></child>
<child href="http://home.netscape.com/escapes/smart_browsing"
name="Learn About Smart Browsing...">
</child>
</RDF:RDF>
</RelatedLinks>
```

كل هذا يحدث خلف الستار أي بدون أن يراه أحد، ولا يعلم المستخدمين أن البيانات قد تم نقلها في XML ويكون العرض الفعلي هو قائمة في Netscape Navigator وليس صفحة XML أو صفحة HTML.

ما تم تقديمه إلى الآن لا يوضح إلا القشور في استخدام XML للبيانات الداخلية. الكثير من المشروعات الأخرى التي تستخدم XML مازالت في البداية والمزيد منهم في طريقهم للبدء في العام المقبل. أغلب تلك المشروعات لن يعلن عنها في الصحافة التجارية ولكن ستقلل تلك المشروعات للشركات التي تنفذها آلاف الدولارات التي تنفقها في التطوير طوال مدة حياة المشروع. يمكن لطبيعة التوثيق الذاتية للغة XML أن تكون مفيدة للبيانات الداخلية لشركة كما هو الحال بالنسبة لبياناتها الخارجية. على سبيل المثال، فمثلا تقوم الشركات الآن بمحاولة لمعرفة ما إذا كان المبرمجين الذين خرجوا للمعاش من ٢٠ سنة قد استخدموا تواريخ ذات رقمين أم لا، إذا كانت هذه هي وظيفتك هل كنت تفضل استخدام بيانات تبدو كما يلي:

3c 79 65 61 72 3e 39 39 3c 2f 79 65 61 72 3e

أو تبدو هكذا:

```
<YEAR>99</YEAR>
```

لسوء الحظ يحاول الكثير من المبرمجين تنظيم البيانات في التنسيق الأول وتجعل لغة XML عملية الوصول إلى الأخطاء أسهل وأسرع في التصحيح.

خلاصة

بدأ هذا الفصل بتوضيح عام للتطبيقات المتنوعة والعديدة التي سيوضع XML من أجلها. بعض هذه التطبيقات مثل CML وMathML وMusicML هي امتداد واضح للغة HTML لمستعرضات الويب، ولكن تطبيقات أخرى مثل OFX وXFDL وHRML تذهب في اتجاهات جديدة تماما.لكل هذه التطبيقات بناء الجمل وsemantics الخاص بها التي يبني عليه XML المضمن. في بعض الحالات، تكون جذور XML واضحة وفي حالات أخرى يمكن قضاء أشهر عديدة تعمل فيه على الرغم مما تسمع من سهولة استخدام XML. استعرضنا في هذا الفصل التطبيقات التالية التي وضعت لغة XML من أجل استخدامها:

- العلوم الجزئية باستخدام لغة CML.
- العلوم والرياضة باستخدام لغة MathML.
- Webcasting مع CDF.
- الأدب الكلاسيكي.
- تعدد الوسائط مع HTML+TIME.
- تحديث البرامج من خلال OSD.
- الرسوم الاتجاهية Vector graphics مع PGML و VML.
- النوتة الموسيقية في MusicML.
- استجابة الصوت الآلي مع VoxML.
- البيانات المالية مع OFX.
- نماذج الربط القانونية مع XFDL.
- معلومات وظيفة الموارد البشرية مع HRM.
- بيانات تفصيلية من خلال RDF.
- XML نفسه متضمنا XSL وXLL و DCD لتحسين XML.
- الاستخدام الداخلي للغة XML من قبل شركات متعددة، بما في ذلك مايكروسوفت و Netscape و Federal Express.
- نبدأ في الفصل التال كتابة مستندات XML الخاص بك وعرضهم في مستعرضات ويب.



الفصل ٣

في هذا الفصل

إنشاء أول مستند
باستخدام XML

يوضح هذا الفصل كيفية إنشاء مستند بسيط باستخدام لغةXML ويكون بالمستند علامـــات يقـوم المستخدم بتعريفها لتتناسب مع المستند. كذلك يشرح هذا الفصل كيفية كتابة ورقة نمط للمســتند لعرض محتويات هذه العلامات ونتعلم في النهاية كيفية تحميل المستندات في مستعرض ويب ليتم عرضها.

يستخدم هذا الفصل العديد من الأمثلة في الشرح إلا أن بعض الحالات الخاصة والاستثنائية لن يتم شرحها في هذا الفصل ولكن سيتم معالجتها فيما يلي من فصول الكتاب. أما بالنسبة للقواعــد الفنية الأساسية فسيتم إتقانها في وقتها لأنه يمكن كما في HTML التعلم وتحقيق الكثير عن طريـق نسخ عدة أمثلة بسيطة أعدها الآخرون وتعديلها لتتاسب احتياجات المستخدم.

لذلك السبب يجب إتباع الأمثلة الموجودة في هذا الفصل بكتابتها وتحميلها في البرامج المختلفة التي يتم توضيحها ويعطي هذا الأسلوب تمهيد مناسب لجعل تفاصيل XML الفنية فـي الفصـول القادمة أسهل في الفهم من خلال سياق هذه الأمثلة المحددة.

مرحباً XML

يتبع هذا الجزء تقليد قديم للمبرمجين عند تقديم لغة جديدة وهذا التقليد هو تقديم تلك اللغـــة مــع برنامج يقوم بطباعة جملة Hello World في المقدمة وعلى الرغم من اعتبار XML لغة ترمـيز وليس لغة برمجة إلا أن هذا المبدأ مازال مطبقاً. من الأفضل عند البدء استخدام مثال كامل يمكن تكبيره بدلا من محاولة البدء بأجزاء أساسية لا تحقق شيئا وحدها دون إضافات فـــإذا واجـهتك مشاكل مع الأدوات الأساسية يسهل تصحيح تلك المشكلات فـــي سـياق المســتندات البسـيطة والقصيرة المستخدمة هنا بدلا من سياق المستندات الأكثر تعقيداً التي سيتم تطويرهـا فـي بقيـة الكتاب.

نتعلم من هذا الجزء كيفية إنشاء مستند XML بسيط وحفظه في ملف وبعد ذلك نلقـي نظـرة علي التعليمات البرمجية وماذا تعني.

إنشاء مستند XML بسيط

يعلمنا هذا الجزء كيفية كتابة مستند XML ونبدأ بأبسط مستند XML يمكن تخيله وهو مدرج فـي تعليمات برمجة ٣-١:

تعليمات برمجة ٣-١: مرحباً XML

```
<?xml version="1.0" standalone="yes"?>
<FOO>
Hello XML!
</FOO>
```

لا يعتبر هذا المستند معقداً ولكنه مستند XML جيد ولنكون أكثر تحديداً فهذا مستند XML جيد التكوين. تستخدم لغة XML مصطلحات خاصة للمستندات التي تعتبرها جيدة وهي تعتمد على مجموعة القواعد التي يقوم المستند باستيفائها وتعتبر كلمة جيد التكوين هي إحدى تلك المصطلحات ويتم مناقشتها فيما بعد. يمكن كتابة هذا المستند في أي محرر نص ملائم مثل المفكرة أو BBEdit أو emacs.



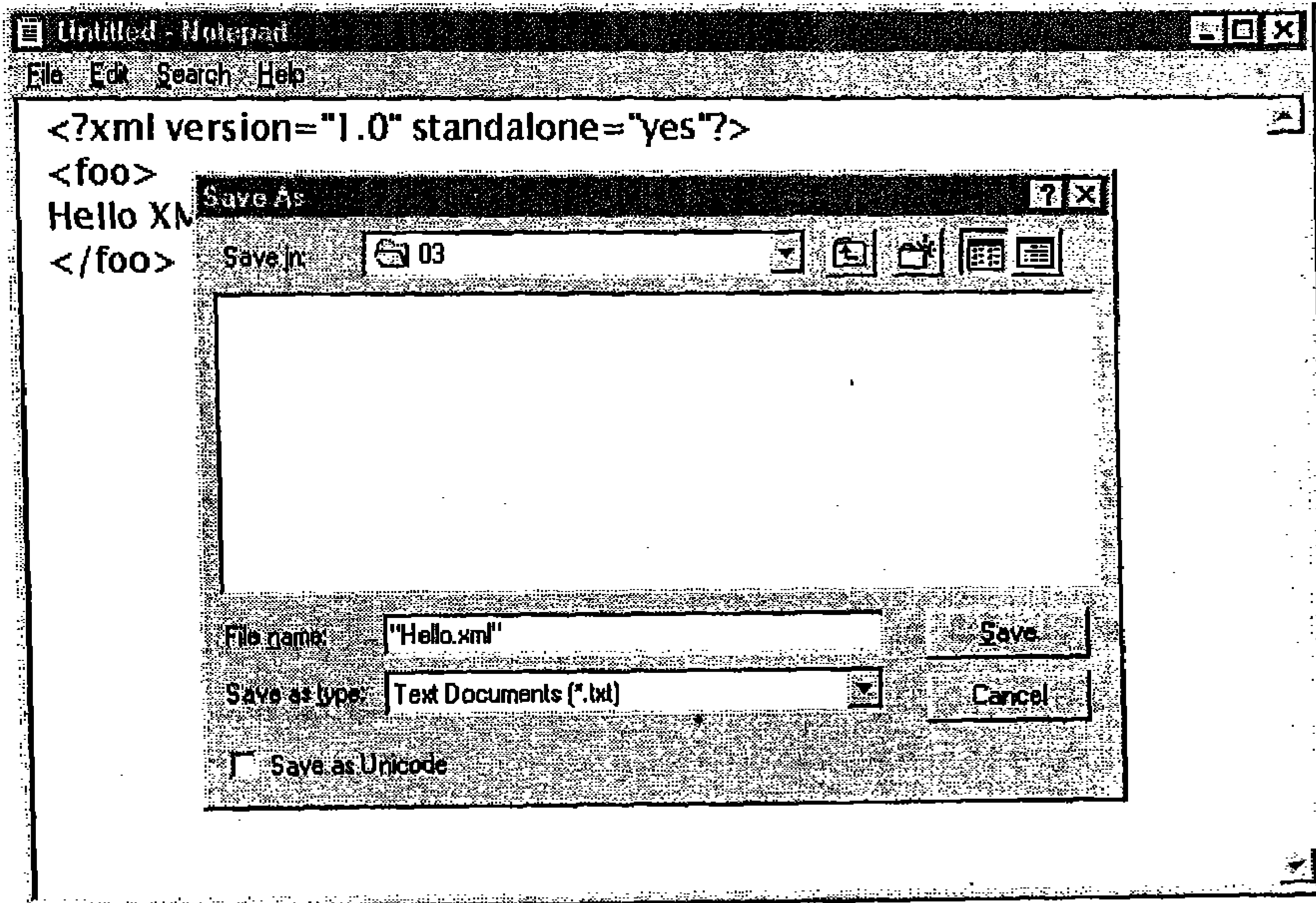
يغطي الفصل ٦ صحة تكوين المستندات "مستندات XML المُنسقة جيداً".

حفظ ملف XML

بعد الانتهاء من كتابة التعليمات البرمجية السابقة احفظ المستند في ملف يسمى HelloWorld.xml, أو MyFirstDocument.xml أو hello.xml أو أي اسم آخر مع الاحتفاظ بملحق اسم الملف xml. في كل الأسماء كملحق قياسي ومع ذلك تأكد من الاحتفاظ به في تنسيق نص عادي وليس في تنسيق أصلي لمعالجة النصوص مثل WordPerfect أو مايكروسوفت وورد.



في حالة استخدام المفكرة لتحرير الملفات في ويندوز ٩٨\٩٥ تأكد عند حفظ المستند من إحاطة اسم الملف بعلامات اقتباس ثنائية مثل "Hello.xml" وليس فقط Hello.xml كما يوضح شكل ٣-١ لأنه بدون علامات الاقتباس تقوم المفكرة بإلحاق الامتداد txt. لاسم الملف ليصبح Hello.xml.txt وهذا ما لا يجب حدوثه على الإطلاق.



الشكل ٣-١ حفظ مستند XML في المفكرة مع إحاطة اسم الملف بعلامات الاقتباس

تتيح المفكرة الخاصة بإصدار Windows NT خيار حفظ الملف في Unicode وعلى الرغم من أن حفظ الملف في Unicode يؤدي الغرض إلا أنه في تلك المرحلة من الأفضل الالتزام بملفات ASCII XML الأساسية والتي قد تكون إما Unicode أو إصدار مضغوط من Unicode يسمى UTF-8 وهو مجموعة قوية من مجموعات ASCII وتعتبر أيضاً ملفات ASCII الخالصة ملفات XML صحيحة.



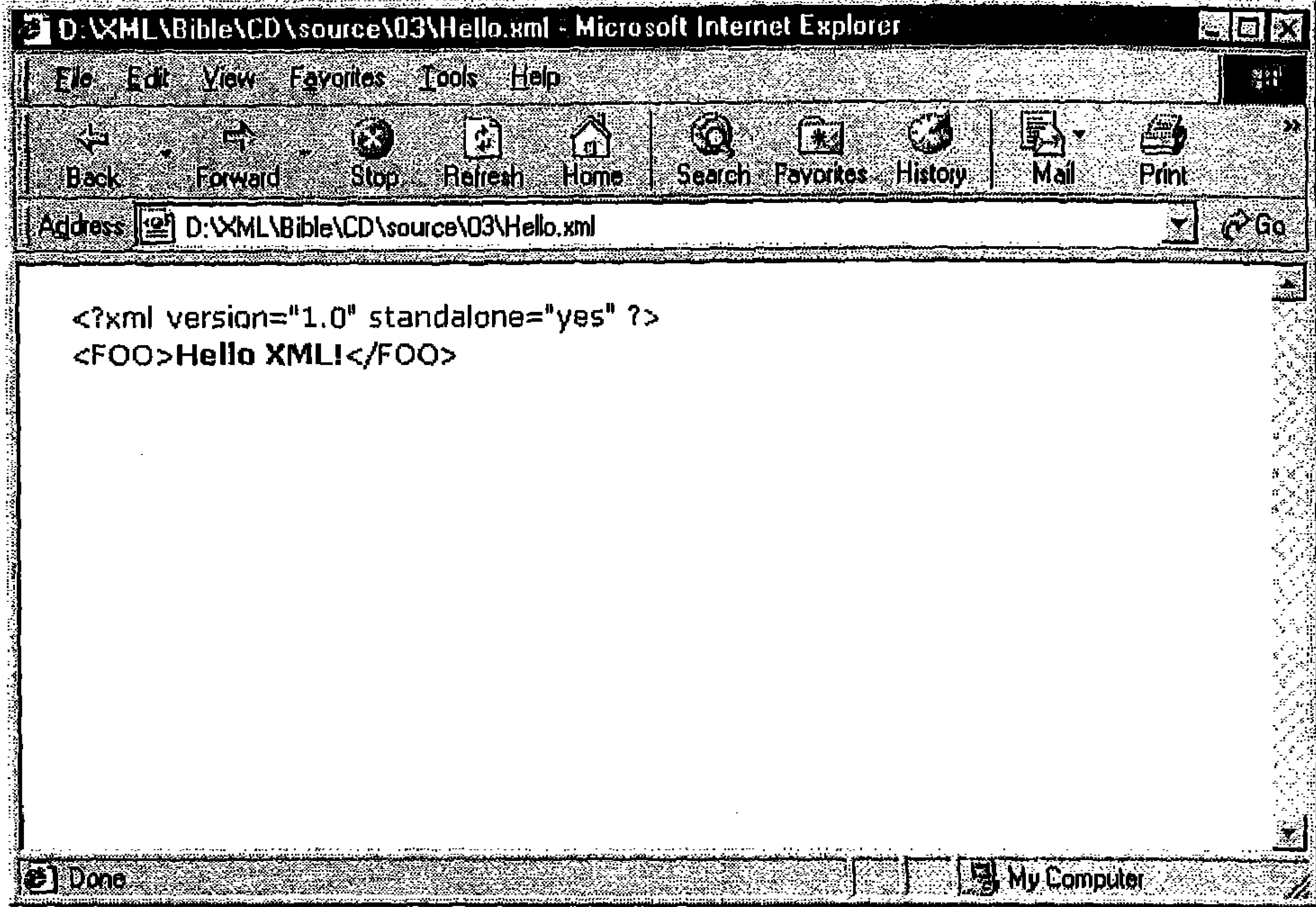
سوف يتم مناقشة ملفات ASCII وUTF-8 بالتفصيل في الفصل ٧ "اللغات الأجنبية والنص غير الروماني".

تحميل ملف XML في مستعرض ويب

يعرض أول مستند تم إنشائه باستخدام لغة XML في مستعرض يدعم XML مثل Internet Explorer5.0 ويوضح الشكل ٣-٢ أول مستند XML.

يختلف شكل المستند الذي يظهر من مستعرض إلى آخر وفي هذا المثال فإن تعليمات مصدر المستند البرمجية منسقة بشكل مناسب وكذلك عرض ألوان بناء الجملة إلا أن هذا لا يعني بالضرورة أن يكون المستند جذابا، وتكمن المشكلة في إن المستعرض لا يعلم ما الذي يجب فعله

مع عنصر FOO ولهذا السبب يجب استخدام ورقة نمط ليعلم المستعرض ما الذي يجب فعله مع كل عنصر ويتم شرح ذلك فيما يلي. أولا لنلقي نظرة على مستند XML الأول الذي تم إنشائه.



الشكل ٣-٢ hello.xml في Internet Explorer 5.0.

استكشاف مستند XML بسيط

نقوم الآن باختبار مستند XML البسيط الموجود في تعليمات برمجة ٣-١ لنفهم جيداً ماذا يعني كل سطر من سطور التعليمات البرمجية. أول سطر هو تعريف XML.

```
<?xml version="1.0" standalone="yes"?>
```

هذا مثال من تعليمات معالجة XML وتبدأ تعليمات المعالجة بالعلامة ?> وتنتهي بالعلامة <? وأول كلمة بعد علامة ?> هي اسم تعليمات المعالجة وفي هذا المثال هي xml.

يكون لتعريف XML سمات هي سمة version وسمة standalone. السمة هي زوج من قيمة اسمية تفصلهما علامة تساوي ويكون الاسم إلى يسار علامة يساوي والقيمة إلى جهة اليمين وتكتب القيمة بين علامات اقتباس مزدوجة.

يبدأ كل مستند XML بتعريف XML الذي يحدد إصدار XML المستخدم. في المثـــال الســـابق تقوم سمة version بتوضيح أن المستند يطابق XML 1.0 وقد يحتوي تعريف XML على ســـمة standalone التي توضح ما إذا كان المستند كاملاً في هذا الملف أم بحاجة إلى استيراد ملفـــات أخرى. بالنسبة لهذا المثال والفصول القادمة تكون المستندات كاملـــة ولـــذا يتـــم إعـــداد ســـمة standalone إلى yes.

بالنسبة للسطور الثلاثة التالية في تعليمات البرمجة ٣-١ وهم:

```
<F00>
Hello XML!
</F00>
```

تمثل تلك السطور الثلاثة مجتمعة عنصر F00 أما في حالة انفصالها عـــن بعضـــها يمثـــل <F00> علامة البدء ويمثل </F00> علامة النهاية ويعتبر Hello XML! هو محتوى عنصـــر F00.

إذا سألت عن معنى علامة <F00>، تكون الإجابة هي أنها تعني أي شئ تريده، فبدلا مـــن الاعتماد على بضع مئات من العلامات السابق تعريفها تتيح لغة XML إنشـــاء العلامـــات التـــي تحتاجها ولهذا فان علامة <F00> تحمل المعنى الذي تقوم بتحديده لها كما يمكن كتابـــة نفـــس مستند XML بأسماء علامات مختلفة كما توضح تعليمات برمجة "٣-٢" و"٣-٣" و"٣-٤" التالية.

تعليمات برمجة ٣-٢ : greeting.xml

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
</GREETING>
```

تعليمات برمجة ٣-٣ : paragraph.xml

```
<?xml version="1.0" standalone="yes"?>
<P>
Hello XML!
</P>
```

تعليمات برمجة ٣-٤ : document.xml

```
<?xml version="1.0" standalone="yes"?>
<DOCUMENT>
Hello XML!
</DOCUMENT>
```

تحتوي مستندات XML الأربعة الموجودة في تعليمات البرمجة من "٣-١" إلى "٣-٤" علامات بأسماء مختلفة، ومع ذلك فكلها متساوية حيث لهم نفس البنية والمحتوى.

تعيين معاني لعلامات XML

يكون لعلامات الترميز ثلاثة أنواع من المعاني وهم البنية والمعنى والنمط. تقسم البنية المستند إلى شجرة عناصر ويقوم المعنى بربط العناصر الفردية بالعالم خارج المستند نفسه ويحدد النمط كيف يتم عرض عنصر.

توضح البنية شكل المستند دون الاهتمام بالاختلافات بين العلامات والعناصر الفردية فمثلا الأربع مستندات XML الموضحة في تعليمات البرمجة من "٣-١" إلى "٣-٤" تتطابق من جهة البنية فكلها تحدد مستندات لعنصر جذري واحد غير فارغ ولهذا فاختلاف أسماء العلامات ليس له أي دلالة هيكلية على بنية المستند.

بالنسبة لمعنى المستند فهو يوجد خارج المستند أي في ذهن المؤلف أو القارئ أو في برنامج كمبيوتر ينتج أو يقرأ تلك الملفات، فمثلا يعين مستعرض ويب الذي يفهم HTML ولا يفهم XML معنى فقرة لعلامات <P> و</P> وليس لعلامات <GREETING> و</GREETING> أو <F00> </F00> أو</DOCUMENT> <DOCUMENT> وكذلك فأي فرد يتحدث اللغة الإنجليزية سيفهم <GREETING> و</GREETING> أو <DOCUMENT> و</DOCUMENT> أكثر من فهمه <F00> و</F00> أو <P> و</P> ولهذا فالمعنى يكون في ذهن المستخدم.

لا يمكننا أن نقول إن أجهزة الكمبيوتر تفهم أي شئ فهي أجهزة تقوم بمعالجة وحدات بت وبايت طبقاً للصيغ التي يتم تحديدها مسبقاً فلا يجد جهاز الكمبيوتر فرق في استخدام </FOO> أو <P> أو استخدام علامات أكثر تفصيلا مثل </GREETING> أو <DOCUMENT> حتى بالنسبة لمستعرض ويب فهو لا يفهم ما معنى فقرة ولكن مستعرضات ويب تعلم أن وجود علامة <P> يعني ترك سطر خالي قبل العنصر التالي.

من الأفضل اختيار علامات تعكس معنى المعلومات التي تتضمنها العلامات وتقوم بعض المؤسسات بالعمل في العديد من المجالات مثل الرياضة والكيمياء على إنشاء مجموعة علامات قياسية يتم استخدامها في الوقت المناسب وعموماً يتم إنشاء معظم العلامات حسب رغبة المستخدم.

ما يلي هو بعض العلامات الممكن استخدامها:

```
<MOLECULE> <INTEGRAL>
<PERSON>    <SALARY>
<author>    <email>
<planet>    <sign>
<Bill>  <plus/>
<Hillary>   <plus/>
<Gennifer>  <plus/>
<Paula>     <plus/>
<Monica>    <equals/>
<divorce>
```

النوع الثالث من المعاني الممكن ربطها بعلامة هو معنى النمط ويحدد معنى النمط كيفية تقديم محتوى العلامة علي شاشة جهاز الكمبيوتر أو أي جهاز مخرجات آخر. أيضاً يحدد معنى النمط ما إذا كان العنصر أسود عريض أو مائل أو أخضر ويكون حجم الخط ٢٤ نقطة أو أي شئ آخر وتفهم أجهزة الكمبيوتر معنى النمط عن فهمها لمعنى المستند المرتبط بالعالم الواقعي ويطبق معنى النمط في XML من خلال أوراق النمط.

كتابة ورقة نمط لمستند XML

تتيح لغة XML إنشاء علامات متعددة وحيث أن للمستخدم الحرية الكاملة في إنشاء العلامات فلا يستطيع أي مستعرض عام توقع العلامات ووضع قواعد لعرضها ولهذا السبب يجب كتابة ورقة نمط لمستند XML ليعلم المستعرض كيفية عرض تلك العلامات ومثلما يحدث في مجموعات العلامات يمكن مشاركة ودمج أوراق النمط بين مستندات مختلفة وأشخاص مختلفين.

كما وضح الفصل الأول تتاح أكثر من لغة لورقة النمط وورقة النمط المستخدمة في هذا الكتاب تسمى Cascading Style Sheets واختصارها هي (CSS) وتملك CSS ميزة كونها معيار W3C القياسي وهي معروفة للكثيرين ممن يستخدمون HTML وأيضا يتم دعمها في أول إنتاج لمستعرضات ويب التي تتيح XML.



كما ذكر في الفصل ١ فهناك لغة ورقة نمط أخرى وهي Extensible Style Language وتعتبر XSL حالياً هي أكثر لغات أوراق النمط مرونة وفاعلية والوحيدة المصممة خصيصا للاستخدام مع XML ومع ذلك فإن XSL أكثر تعقيداً من CSS وهي لم يتم دعمها بنفس درجة دعم لغة CSS وكذلك لم يتم الانتهاء منها بعد.



سوف يتم شرح XSL في الفصول ٥ و١٤ و١٥.

يحتوي المثال الموضح في تعليمات برمجة "٣-٢" greeting.xml على علامة واحدة وهي <GREETING> ولذا يجب تعريف النمط لعنصر GREETING. أما تعليمات برمجة "٣-٥" فهي ورقة نمط بسيطة تحدد محتويات عنصر GREETING التي يجب تقديمها كعنصر مكون من وحدة واحدة ويكون نوع الخط أسود عريض وحجمه ٢٤ نقطة.

تعليمات برمجة ٣-٥ : greeting.xsl

```
GREETING {display: block; font-size: 24pt; font-weight: bold;
```

يجب كتابة تعليمات برمجة "٣-٥" في محرر نص وحفظها في ملف جديد يسمى greeting.css في نفس دليل تعليمات برمجة "٣-٢" ويرمز ملحق الملف css إلى Cascading Style Sheet. يعتبر ملحق الملف css مهماً على الرغم من عدم أهمية اسم الملف نفسه ومع ذلك فإذا تم تطبيق ورقة النمط على مستند XML واحد فمن الأفضل إعطاء ورقة النمط نفس اسم المستند مع إبدال اسم ملحق ملف ورقة النمط إلى CSS بدلا من xml.

إرفاق ورق نمط بمستند XML

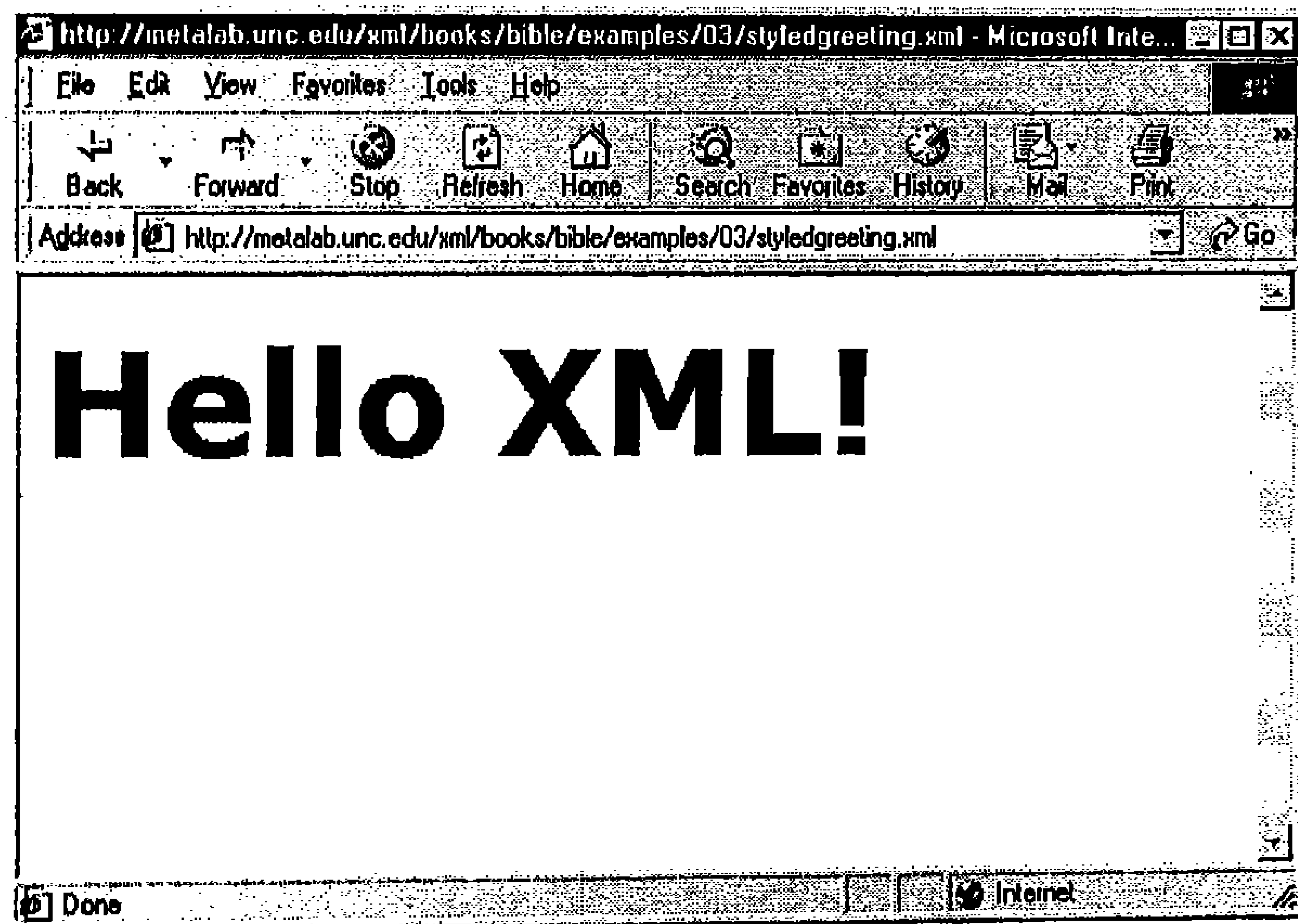
بعد الانتهاء من كتابة مستند XML وورقة نمط CSS لهذا المستند يجب إعلام المستعرض لتطبيق ورقة النمط على المستند ويوجد عدد من الأساليب المختلفة لتنفيذ ذلك بما في ذلك الاتصال بين الخادم والمستعرض عبر رؤوس صفحة HTTP واصطلاحات التسمية وافتراضات جهة المستعرض. الأسلوب الوحيد المستخدم حالياً هو أسلوب تضمين تعليمات معالجة أخرى في مستند XML لتحديد ورقة النمط المستخدمة.

تعليمات المعالجة هي <?xml-stylesheet?> ولها سمتان هما: type وherf. تحدد سمة type لغة ورقة النمط المستخدمة وتحدد سمة href محدد موقع، قد يكون مرتبط بالمكان الذي توجد فيه ورقة النمط. تحدد تعليمات معالجة xml-stylesheet الموجودة في تعليمات برمجة "٣-٦" وجوب تطبيق ورقة النمط المسماة greeting css المكتوبة بلغة ورقة نمط CSS على هذا المستند.

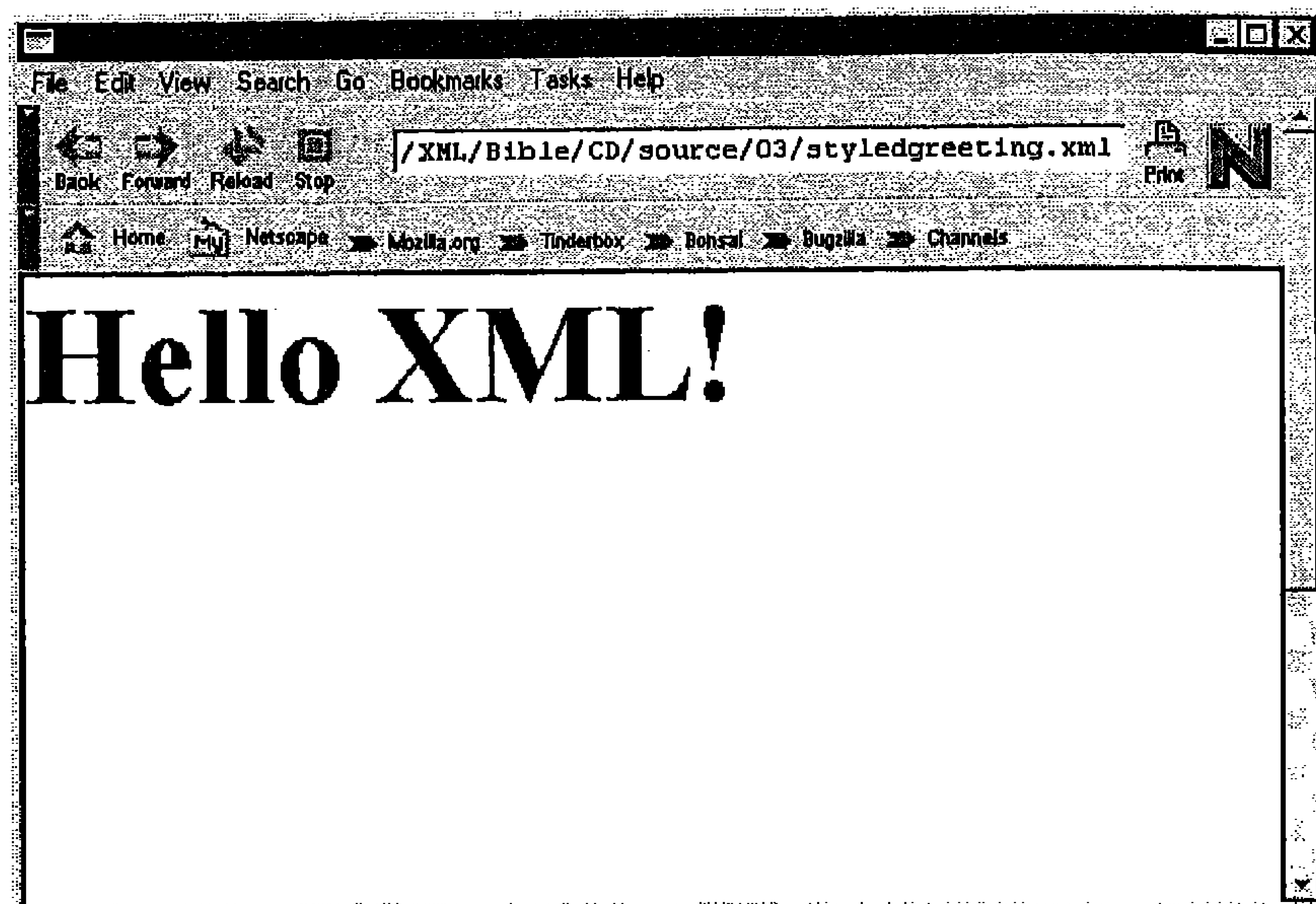
تعليمات برمجة ٣-٦: ملف styledgreeting.xml مرتبط بتعليمات معالجة موجودة في ورقة نمط xml

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css2" href="greeting.css"?>
<GREETING>
Hello XML!
</GREETING>
```

لرؤية أول مستند xml تم إنشائه وورقة النمط الخاصة به يجب تحميل تعليمات برمجة "٣-٦" في مستعرض Mozilla أو Internet Explorer 5.0. يوضح الشكل "٣-٣" styledgreeting في Internet Explorer 5.0 ويوضح الشكل "٣-٤" styledgreeting. في بناء متطور حديث من Mozilla.



الشكل ٣-٣ Styledgreeting.xml في Internet Explorer 5.0.



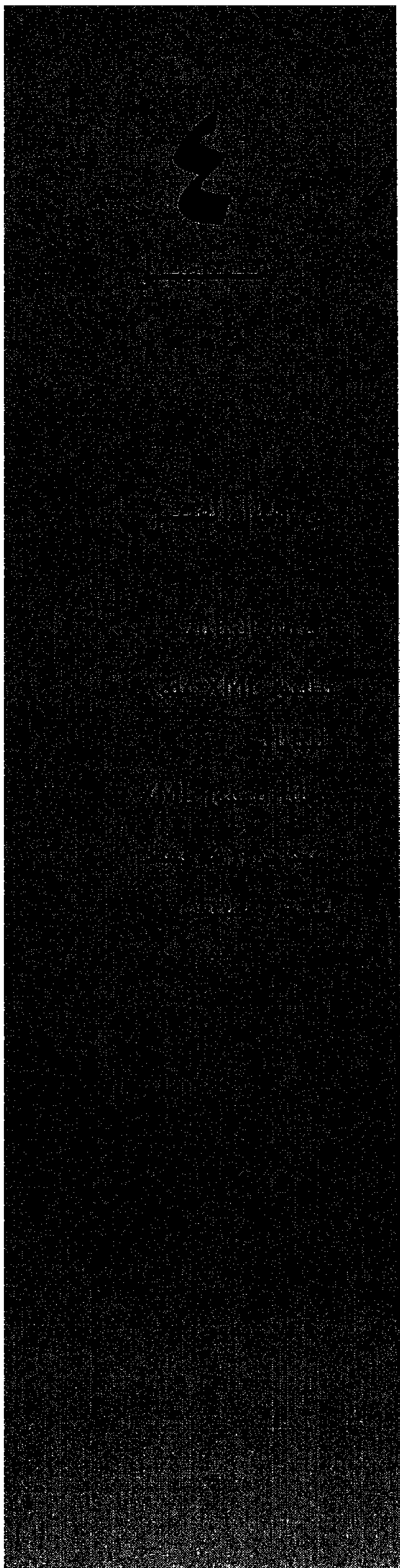
الشكل ٣-٤ Styledgreeting.xml في بناء متطور حديث من Mozilla.

خلاصة

تعلمت في هذا الفصل كيفية إنشاء مستند XML بسيط وكذلك تعلمت:

- كيفية كتابة وحفظ مستندات XML بسيطة.
- كيفية تعيين المعاني الثلاثة لعلامات XML وتلك المعاني هي: البنية والمعنى والنمط.
- كيفية كتابة ورقة نمط CSS لمستند XML التي تعّف المستعرض كيفية عرض علامات محددة.
- كيفية إرفاق ورقة نمط CSS بمستند XML مع تعليمات معالجة xml-stylesheet.
- كيفية تحميل مستندات XML في مستعرض ويب.

نوضح في الفصل القادم مثال لمستند XML يوضح الاعتبارات العملية المتدخلة في اختيار علامات XML.



هيكلة البيانات

يوضح هذا الفصل عن طريق مثال واحد كيفية حفظ قائمة كبيرة من إحصاءات رياضة البيسبول "لعبة كرة القدم الأمريكية" أو أي بيانات مشابهة في XML ويكون لمثل هذا المستند عدة استخدامات محتملة وأكثر تلك الاستخدامات شيوعاً هي وضع المستند على صفحة ويب أو استخدامه كإدخالات لبرامج أخرى لتحليل مواسم رياضة أو قوائم معينة. كما يوضح هذا الفصل كيفية ترميز البيانات في XML ولماذا يتم اختيار علامات XML وكيفية إعداد ورقة نمط CSS للمستند.

اختبار البيانات

في أثناء كتابة هذا الكتاب أكتوبر ١٩٩٨ استطاع فريق New York Yankees الفوز بالبطولة رقم ٢٤ World Series بعد هزيمة فريق San Diego Padres في أربعة مباريات ولقد أنهى فريق Yankees الموسم العادي بتسجيل ١١٤ فوز في قائمة بطولة الدوري الأمريكي ويعتبر عام ١٩٩٨ موسم مذهل بالنسبة لرياضة البيسبول عامة فقد تنافس كل من Mark McGwire لاعب فريق St. Louis Cardinals واللاعب Sammy Sosa لاعب فريق Chicago Cubs خلال شهر سبتمبر في محاولة لتصدر قائمة home runs التي يتصدرها Roger Maris. ومازال يتصدر قائمة الاتحاد الرئيسية في home runs لموسم واحد اللاعب Josh Gibson والذي حقق ٧٥ هدف home runs في دوري Negro عام ١٩٣١ وفي الواقع لم يكن على Gibson مواجهة أنواع الضربات التي واجهها Sosa وMcGwire في الدوري الموحد اليوم وأيضاً لم يواجه Babe Ruth الذي كان يعتقد وبصورة غير صحيحة أنه كان على قمة القائمة إلى إن أحرز Roger Maris ٦١ هدف عام ١٩٦١.

ما الأسباب التي جعلت موسم ١٩٩٨ موسم مثير؟ قد تكون الإجابة على هذا السؤال هي أن موسم ١٩٩٨ كان عام موسع بمعني أن عدد الفرق زاد ٣ فرق جديدة وبالتالي ضعف قوة رمي الكرة عموماً وهذا يعطي ضاربي كرة بارزين مثل Sosa وmcGwire وفريق متميز مثل Yankees الفرصة للظهور على الرغم من أنهم بمثل قوتهم التي كانت عام ١٩٩٧ وذلك لأن متوسط المنافسين الذين واجهوهم كان أضعف بكثير ولكن السبب الأساسي الذي جعل موسم ١٩٩٨ موسم مثير هو الإحصاءات.

المذهل في هذه اللعبة هو اختلافها عن بقية الألعاب حيث أن الأساسي في المنافسة ليس كما في بقية الألعاب الجرأة والشجاعة والمهارة الخ ولكن هو الأرقام: أي متوسط عدد مرات ضرب الكرة ومتوسط عدد مرات ضرب الكرة لمستخدمي اليد اليمني في قذف الكرة ولمستخدمي اليد اليسرى في قذف الكرة ونسبة الهجوم الخ.

يتركز اهتمام مشجعي لعبة البيسبول على الأرقام كلما زادت الأرقام كلما كان أفضل وتكون الإنترنت في كل موسم مليئة بالآلاف من الاتحادات rotisserie حيث تتم إدارة الفرق وتبادل

اللاعبين وحساب جودة أداء اللاعبين اعتماداً على الأداء الفعلي والواقعي للاعبين. تتتبع STATS, Inc نتائج كل رمية كرة تحدث في مباريات الدوري الرئيسي ولذا فمن السهل استنتاج أن أحد ضاربي الكرة أداءه أفضل من متوسط أداء الآخرين في مراكز التسجيل أو أسوء.

في الأجزاء التي تلي ذلك نقوم باختبار الإحصاءات الشائعة والتي تصف ضرب الكرة ورمي الكرة بالنسبة للاعب ويتم تجاهل إحصاءات الوسط "الهجوم" لنجعل حجم المثال معقول ليمكن إدارته والتعامل معه والمثال المستخدم هنا هو لفريق New York Yankees.

ضارب الكرة Batters

منذ عدة سنوات تم كتابه بحث بعنوان:

"A Markov Chain Approach to Baseball" (Operations Research, Volume 45, Number 1, January-February, 1997, pp. 14-23, http://www.math.njit.edu/~bukiet/Papers/ball.pdf)

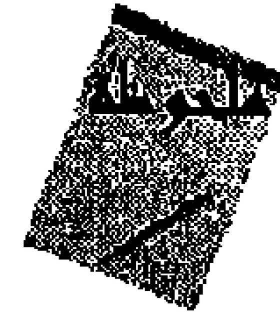
حيث تم تحليل كل ترتيبات ضرب الكره الممكنة لكل الفرق في بطولة الدوري العام ١٩٨٩ وكانت نتائج هذا البحث مثيرة حيث أظهرت مثلا إن أسوء ضارب للكرة في الفريق وغالباً ما يكون رامي كرة البيسبول يجب أن يضرب الكرة من المركز الثامن بدلا من المركز التاسع ولكن ما نحن بصدد مناقشته هو الإجراءات التي اتخذت لإظهار هذا البحث. كانت مهمتي هي إعادة إدخال البيانات الكاملة لتاريخ ضرب الكرة لكل لاعب في بطولة الدوري العام يدوياً وكانت بيانات كل لاعب تقدم في صفوف أرقام كما يوضح جدول ٤-١: ضاربي كرة فريق YanKees لعام ١٩٩٨، وقد تم فقط إدراج اللاعبين الذين قاموا بضرب الكرة فعلاً حيث أن رامي الكرة نادر ما يضرب الكرة في بطولة الدوري الأمريكي.

يقوم كل عمود بتعريف عنصر محدد ولذا يجب وجود عناصر للاعب والمركز وعدد اللعبات التي لعبت والضربات الصحيحة للكرة وعدد اللفات "الدورات" والأهداف وعدد الرميات الثنائية والرميات الثلاثية والدورات الكاملة وغالباً لا يتم تسجيل الرميات الفردية بصورة منفردة ولكن يتم احتسابهم عن طريق طرح مجموع الرميات الثنائية والثلاثية والدورات الكاملة من عدد الأهداف.

جدول ٤–١

ضاربو كرة فريق YanKees عام ١٩٩٨

Name	Position	Games Played	At Bats	Runs	Hits	Doubles	Triples	Home Runs	Runs Batted In	Strike Walks	Outs	Hit by Pitch
Scott Brosius	Third Base	152	530	86	159	34	0	19	98	52	97	10
Homer Bush	Second Base	45	71	17	27	3	0	1	5	5	19	0
Chad Curtis	Outfield	151	456	79	111	21	1	10	56	75	80	7
Chili Davis	Designaed Hitter	35	103	11	30	7	0	3	9	14	18	0
Mike Figga	Catcher	1	4	1	1	0	0	0	0	0	1	0
Joe Giradi	Catcher	78	254	31	70	11	4	3	31	14	38	2
Derek Jeter	Shortstop	149	626	127	203	25	8	19	84	57	119	5
Chuck Knoblauch	Second Base	150 18	603	117	160	25	4	17	64	76	70	
Ricky Ledee	Outfield	42	79	13	19	5	2	1	12	7	29	0
Mike Lowell	Third Base	8	15	1	4	0	0	0	0	0	1	0
Tino Martinez	First Base	142	531	92	149	33	1	28	123	61	83	6
Paul	Outfield	152	602	95	191	40	2	24	116	57	103	2
Jorgen Posada	Catcher	111	358	56	96	23	0	17	63	47	92	0
Tim Raines	Outfield	109	321	53	93	13	1	5	47	55	49	3
Luis Sojo	Shortstop	54	147	16	34	3	1	0	14	4	15	0
Shane Spencer	Outfield	27	67	18	25	6	0	10	27	5	12	0
Darryl Strawberry	Designated Hitter	101	295	44	73	11	2	24	57	46	90	3
Dale Sveum	First Base	30	58	6	9	0	0	0	3	4	16	0
Bernie Williams	Outfield	128	499	101	169	30	5	26		74	81	1



تعتبر البيانات في الجدول السابق وبيانات رامي الكرة في الجزء التالي قائمة محددة نوعاً ما حيث تحدد البيانات التي يتم تجميعها في لعب بيسبول عادية وهناك عناصر أخرى متعددة ولكن سنلتزم في هذا المثال بتلك المعلومات الأساسية لنستطيع التعامل مع المثال.

رامي الكرة Pitchers

لا يتوقع من رامي الكرة أن يكون هداف home-run أو أن يكون base-stealers وبالطبع يعتبر رامي الكرة الذي يستطيع الوصول إلى الهدف أولا عند الضرورة إضافة للفريق، ولكن يتم الحكم على رامي الكرة بمجموعة أرقام مختلفة تماما يوضحها جدول ٤-٢. يقوم كل عمود في الجدول بتعريف عنصر وبعض هذه العناصر مثل الاسم والمركز هي نفس العناصر المستخدمة بالنسبة لرامي الكرة وضارب الكرة أما بالنسبة للبعض الأخر مثل عناصر منع تسجيل الأهداف أو التعادلات تطبق فقط على رامي الكرة وهناك بعض العناصر التي لها نفس الاسم كما في إحصاءات ضارب الكرة إلا أن لها معني أخر مثل runs وhome runs.على سبيل المثال عدد runs لضارب الكرة هو عدد runs التي أحرزها ضارب الكرة أما بالنسبة لرامي الكرة فعدد runs هو عدد runs التي أحرزها الفريق المنافس ضد رامي الكرة.

تنظيم بيانات XML

يستند XML على نموذج احتواء وهذا يعني أن يحتوي كل عنصر XML على نص أو عناصر XML أخرى تسمي عناصر فرعية وقد تحتوي بعض عناصر XML على كلا من النص والعناصر الفرعية ويعتبر هذا نموذج غير جيد يجب تجنبه كلما أمكن ذلك.

هناك عدة طرق لتنظيم البيانات اعتماداً على احتياجات المستخدم، فإحدى مزايا XML أنه يجعل كتابة برنامج يعيد تنظيم البيانات في نموذج مختلف عملية مباشرة جداً ويتم مناقشة ذلك عند مناقشة تحولات XSL في الفصل ١٤.

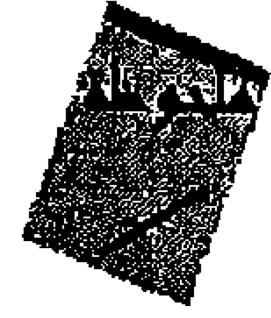
لنستطيع البدء في تنظيم بيانات XML يجب الإجابة على سؤال هو: ما الذي يحتوي على ماذا؟ على سبيل المثال تحتوي بطولة الدوري على مجموعات وتحتوي تلك المجموعات على فرق وتحتوي الفرق على لاعبين، وعلى الرغم من أن الفريق يستطيع تغيير المجموعات عند الانتقال من مدينة إلى أخرى وكذلك تبادل اللاعبين في أي وقت إلا أن كل لاعب ينتمي لفريق واحد محدد وينتمي كل فريق لمجموعة واحدة محددة أيضاً. كذلك يحتوي الموسم على مباريات تحتوي تلك المباريات على جولات وتحتوي الجولات على at bats وتحتوي تلك على رميات للكرة أو لعبات.

هل يحتوي الموسم على بطولة دوري أم هل تحتوي بطولة الدوري على موسم؟ لا توجد إجابة واضحة وواحدة لهذا السؤال حيث تعتمد إمكانية جعل عناصر الموسم عناصر تابعة لعناصر بطولة الدوري أو جعل عناصر بطولة الدوري عناصر تابعة لعناصر الموسم على استخدام البيانات التي يتم تنظيمها. يمكن أيضاً إنشاء عنصر جذري جديد يحتوي على كلا من المواسم وبطولات الدوري بدون أن يكون أحدهما عنصر تابع للأخر على الرغم من أن تنفيذ ذلك يتطلب تقنيات متقدمة لن يتم مناقشتها الآن وعلى الرغم من ذلك يمكن تنظيم البيانات كما يرغب المستخدم.

جدول ٤-٢

رامي كرة فريق Yan-Kees ١٩٩٨

Name	Position	W	L	S	G	GS	CG	SHO	ERA	IP	H	HR	R	ER	HB	WP	BK	WB	SO
Joe Borowski	Relief Pitcher	1	0	0	8	0	0	0	6.52	9.2	11	0	7	7	0	0	0	4	7
Ryan Bradley	Relief Pitcher	2	1	0	5	1	0	0	568	12.2	12	2	9	8	1	0	0	9	13
Jim Bruske	Relief Pitcher	1	0	0	3	1	0	0	3	9	9	2	3	3	0	0	0	1	[illegible]
Mike Buddie	Relief Pitcher	4	1	0	24	2	0	0	5.62	41.2	46	5	29	26	3	2	1	13	20
David Cone	Starting Pitcher	20	7	0	31	31	3	0	3.55	207.2	186	20	89	82	15	6	0	59	209
Todd Erdos	Relief Pitcher	0	0	0	2	0	0	0	9	2	5	0	2	2	0	0	0	1	0
Orlando Holmes	Starting Pitcher	12	4	0	21	21	3	1	3.13	141	113	11	53	49	6	5	2	52	131
Darren Holmes	Relief Pitcher	0	3	2	34	0	0	0	3.33	51.1	53	4	19	19	2	1	0	14	31
Hideki Irabu	Starting Pitcher	13	9	0	29	28	2	1	4.06	173	148	27	79	78	9	6	1	76	126
Mike Jerzembeck	Starting Pitcher	0	1	0	3	2	0	0	12.79	6.1	9	2	9	9	0	1	1	4	1
Graeme Lloyd	Relief Pitcher	3	0	0	50	0	0	0	1.67	37.2	26	3	10	7	2	2	0	6	20
Ramiro Mendoza	Relief Pitcher	10	2	1	41	14	1	1	3.25	130.1	131	9	50	47	9	3	0	30	56
Jeff Nelson	Relief Pitcher	5	3	3	45	0	0	0	3.79	40.1	44	1	18	17	8	2	0	22	35
Andy Pettitte	Starting Pitcher	16	11	0	33	32	5	0	4.24	216.1	226	201	101	2	6	5	0	87	146
Mariano Rivera	Relief Pitcher	3	0	36	54	0	0	0	1.91	61.1	48	3	13	13	1	0	0	17	36
Mike Stanton	Relief Pitcher	4	1	6	67	0	0	0	5.47	79	71	13	51	48	4	0	0	26	69
Jay Tessmer	Relief Pitcher	1	0	0	7	0	0	0	3.12	8.2	4	1	3	3	0	1	0	4	6
David Wells	Starting Pitcher	18	4	0	30	30	8	5	3.49	214.1	195	29	86	83	1	2	0	29	163



من يعلم نظرية قاعدة البيانات بتعرف على نموذج XML على أساس إنها قـــاعدة بيانات هيكلية وبالتالية فهي تتشارك مع نموذج البيانات في كل العيـــوب وبعـــض المزايا. يكون في بعض الأوقات اتجاه العلاقات التي تعتمد على الجـــداول أكـــثر فاعلية والمثال الذي نستخدمه يتفق مع هذا الاتجاه ولكن XML لا ينتـــج نمـــوذج علائقي.

من جهة أخرى يمكن تخزين البيانات الفعلية في جداول متعددة في قاعدة بيانـــات علائقية ويلي ذلك إنشاء مستند XML بسرعة وبالطبع فقـــد تـــم إنشـــاء الأمثلـــة الموجودة على الأقراص المضغوطة بهذا الأسلوب وهذا يتيـــح تقديـــم مجموعـــة واحدة من البيانات في تنسيقات متعددة ويوفر نقل البيانات باستخدام أوراق النمـــط طريقة عرض أخرى للبيانات.

حيث أن الهدف الأساسي في المثال هو تحليل أداء اللاعبين في موسم واحد، سنجعل الموســـم هو جذر "أساس" المستندات ويحتوي كل موسم على بطولات دوري وتحتوي بطولات الـــدوري على مجموعات وتحتوي المجموعات على لاعبين ولكن لن نسهب في سرد البيانات إلى مســتوى اللعبات الفردية والجولات لان ذلك سيجعل المثال طويلا أكثر مما ينبغي.

يمكن أيضاً تقسيم البيانات بأي طريقه أخرى تخدم هدف المستخدم فهناك دائماً أكثر من وسيلة لتنظيم البيانات في XML وسنرجع إلى هذا المثال في الفصول القادمة عند استكشـــاف مفـــردات تميز بديلة.

تطبيق XML على البيانات

نبدأ عملية ترميز بيانات موسم بطولة الدوري الرئيسي لعام ١٩٩٨ باستخدام علامات XML التـــي يقوم المبرمج بتعريفها وتذكر أنه يمكن إنشاء العلامات في XML أثناء العمل. سنجعل العنصـــر الأساسي للمستند هو الموسم وتحتوي المواسم على بطولات للدوري وتحتوي بطولات الـــدوري على مجموعات وتتضمن المجموعات فرق والفرق تشمل لاعبين ويكون للاعبيـــن إحصـــاءات تشتمل على المباريات التي لعبت وat bats وruns وhits وdoubles وtriples وhome runs وruns batted in وwalks وhits by pitch.

بدء المستند: تصريح XML والعنصر الجذري

يتم التعرف على مستندات XML بواسطة تصاريح XML وهذه تعليمات معالجة توضع في بداية كل ملفات XML التي تقوم بتعريف الإصدار المستخدم والإصدار الوحيد المفهوم حاليا هو الإصدار ١,٠.

```
<?xml version="1.0"?>
```

لابد أن يحصل كل مستند XML جيد "يتم مناقشة معني كلمة جيد في الفصل التالي" على عنصر جذري وهو عنصر يحتوي على كل العناصر الأخرى للمستند. تجئ علامة العنصر الجذري قبل علامات بدء العناصر الأخرى وتأتي علامة نهاية العنصر الجذري بعد كل علامات نهاية العناصر الأخرى. بالنسبة للعنصر الجذري في المثال نستخدم SEASON بعلامة بدء<SEASON> وعلامة نهاية </SEASON> ويبدو المستند كالتالي:

```
<?xml version="1.0"?>
<SEASON>
</SEASON>
```

لا يعتبر تصريح XML عنصر أو علامة بل هو تعليمات معالجة ولهذا فلا يجب تضمينه داخل العنصر الجذري SEASON ولكن كل عنصر يتم وضعة في هذا المستند سيوضع بين علامة البدء <SEASON> وعلامة النهاية </SEASON>.

يعني اختيار هذا العنصر الجذري انه لن يمكن تخزين مواسم متعددة في ملف واحد وإذا أردنا فعل ذلك يمكن تعريف عنصر جذري جديد يحتوي على مواسم مثل:

```
<?xml version="1.0"?>
<DOCUMENT>
  <SEASON>
  </SEASON>
  <SEASON>
  </SEASON>
</DOCUMENT>
```

اصطلاحات التسمية

يجب إن نعلم قبل البدء في العمل باستخدام XML أن أسماء عناصر XML تمتاز بالمرونة ويمكن أن تحتوي على أي عدد من الأحرف والأرقام إما بالحروف اللاتينية الكبيرة أو الصغيرة لذلك يملك المبرمج حرية اختيار كتابة علامات XML والتي تبدو كما يلي:

اصطلاحات التسمية

<SEASON>
<Season>
<season>
<season1998>
<Season98>
<season_98>

هناك عدة آلاف أخرى من طرق الكتابة المختلفة وعامة لا يهتم XML بأسلوب الكتابة سواء استخدم الأحرف اللاتينية الكبيرة أو الصغيرة أو خليط من الاثنين معاً إلا انه من المفضل استخدام اصطلاح واحد فقط والالتزام به.

بالطبع يجب تعريف الموسم الذي نحلله ولفعل ذلك يمكن إعطاء عنصر SEASON عنصر تابع يحدد YEAR مثل:

```
<?xml version="1.0"?>
<SEASON>
 <YEAR>
  1998
 </YEAR>
</SEASON>
```

تم استخدام المسافة البادئة في هذا المثال وعدة أمثلة أخرى لتشير إلى أن عنصر YEAR هو عنصر تابع لعنصر SEASON وأن النص ١٩٩٨ هو ما يحتوي عليه عنصر YEAR ويعتبر هذا نمط تعليمات برمجية جيد ولكن غير مطلوب لان المساحة الفارغة في XML لا ترمز إلى شيء معين وكان من الممكن كتابة نفس المثال بالأسلوب التالي:

```
<?xml version="1.0"?>
<SEASON>
 <YEAR>1998</YEAR>
</SEASON>
```

يمكن ضغط العناصر في سطر واحد عندما يكون ذلك ملائما ويمكن أيضاً ضغط المستند كله إلى خط ولكن بذلك تفتقد الوضوح في المستند ومثال لذلك:

```
<?xml version="1.0"?><SEASON><YEAR>1998</YEAR></SEASON>
```

بالطبع هذه الطريقة أصعب في القراءة والفهم ولذلك لم تتم الكتابة بهذا الأسلوب. الهدف العاشر المدرج في مواصفات XML 1.0 هو أن الإيجاز في لغة ترميز XML ليس ذو أهمية قصوى ويعكس مثال البيسبول هذا الهدف بوضوح.

تطبيق XML على بيانات بطولة الدوري والمجموعات والفرق

ينقسم الدوري الرئيسي في البيسبول إلى بطولتين هما بطولة الدوري الأمريكي وبطولة الدوري العام ولكل دوري اسمه ويمكن كتابة الاسمين بالرموز كما يلي:

```
<?xml version="1.0"?>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>National League</LEAGUE_NAME>
  </LEAGUE>
  <LEAGUE>
    <LEAGUE_NAME>American League</LEAGUE_NAME>
  </LEAGUE>
</SEASON>
```

تم تعريف اسم الدوري بعنصر LEAGUE_NAME بدلا من عنصر NAME فقط لان عنصر NAME عنصر عام يمكن استخدامه في أي سباق أخر وعلى سبيل المثال فلكل من المجموعات والفرق وكذلك اللاعبين أسماء أيضاً.



يمكن دمج العناصر التي لها نفس الاسم في مجالات مختلفة باستخدام حيز الاسم Namespaces والتي سيتم مناقشتها تفصيليا في الفصل ١٨ ولا يفضل عند استخدام حيز الاسم إعطاء عناصر متعددة في نفس المجال نفس الاسم وفي هذا المثال: TEAM وLEAGUE.

يمكن تقسيم كل دوري إلى مجموعات شرق وغرب ووسط أمريكا والتي يمكن وضعها في شكل رموز كما يلي:

```
<LEAGUE>
  <LEAGUE_NAME>National League</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
  </DIVISION>
```

```
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American League</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
  </DIVISION>
</LEAGUE>
```

تعتمد القيمة الصحيحة للعنصر على أصل العنصر. أصل العنصر هو العناصر التي يحتويها بالإضافة إلى نفسه ولكل من بطولة الدوري العام وبطولة الدوري الأمريكي مجموعة شرقية "East" ولكنهم ليس نفس الشيء.

تنقسم كل مجموعة إلى فرق ولكل فريق اسم ومدينة فمثلاً يمكن وضع البيانات التي تخص بطولة الدوري الأمريكي المجموعة الشرقية في رموز كما يلي:

```
<DIVISION>
  <DIVISION_NAME>East</DIVISION_NAME>
  <TEAM>
    <TEAM_CITY>Baltimore</TEAM_CITY>
    <TEAM_NAME>Orioles</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Boston</TEAM_CITY>
    <TEAM_NAME>Red Sox</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>New York</TEAM_CITY>
```

```
    <TEAM_NAME>Yankees</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Tampa Bay</TEAM_CITY>
    <TEAM_NAME>Devil Rays</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Toronto</TEAM_CITY>
    <TEAM_NAME>Blue Jays</TEAM_NAME>
  </TEAM>
</DIVISION>
```

Each divisi

تطبيق XML على بيانات اللاعب

يتكون كل فريق من لاعبين ولكل لاعب اسم أول "اسم الشخص" ولقب ومن المهم فصل الاســـم الأول عن اللقب حتى يتم الفرز بأحدهما ويمكن وضع بيانات رماة الكرة الجدد في صفوف فريـق Yankees عام ١٩٩٨ في رموز كما يلي:

```
<TEAM>
  <TEAM_CITY>New York</TEAM_CITY>
  <TEAM_NAME>Yankees</TEAM_NAME>
  <PLAYER>
    <GIVEN_NAME>Orlando</GIVEN_NAME>
    <SURNAME>Hernandez</SURNAME>
  </PLAYER>
  <PLAYER>
    <GIVEN_NAME>David</GIVEN_NAME>
    <SURNAME>Cone</SURNAME>

  </PLAYER>
  <PLAYER>
    <GIVEN_NAME>David</GIVEN_NAME>
    <SURNAME>Wells</SURNAME>
  </PLAYER>
```

```
<PLAYER>
  <GIVEN_NAME>Andy</GIVEN_NAME>
  <SURNAME>Pettitte</SURNAME>
 </PLAYER>
 <PLAYER>
  <GIVEN_NAME>Hideki</GIVEN_NAME>
  <SURNAME>Irabu</SURNAME>
 </PLAYER>
</TEAM>
```



يفضل استخدام علامات <GIVEN_NAME> و<SURNAME> عن استخدام علامات <FIRST_NAME>و <LAST_NAME> و<FIRST_NAME> أو <FAMILY_NAME> لكونها أكثر وضوحاً ويعتمد مجيء اسم العائلة "اللقب" أو اسم الشخص نفسه في البداية أو في النهاية طبقاً لحضارة وثقافة الدولة وبالإضافة إلى ذلك فالاسم الأخير ليس بالضرورة اسم العائلة في كل الثقافات.

تطبيق XML على إحصاءات اللاعب

الخطوة التالية هي إحصاءات لكل لاعب وتبدو الإحصاءات مختلفة بعض الشيء بالنسبة لرامي الكرة وضارب الكرة خاصة في الدوري الأمريكي حيث يقوم بعض رماة الكرة بضرب الكرة. يوضح المثال أسفل الكلام إحصاءات عام ١٩٩٨ للاعب Joe Girardi وهو الذي يقوم بالتقاط الكرة catcher ولذا يتم استخدام إحصاءات ضرب الكرة:

```
<PLAYER>
<GIVEN_NAME>Joe </GIVEN_NAME>
  <SURNAME>Girardi</SURNAME>
  <POSITION>Catcher</POSITION>
  <GAMES>78</GAMES>
  <GAMES_STARTED>76</GAMES_STARTED>
  <AT_BATS>254</AT_BATS>
  <RUNS>31</RUNS>
  <HITS>70</HITS>
  <DOUBLES>11</DOUBLES>
  <TRIPLES>4</TRIPLES>
```

```
  <HOME_RUNS>3</HOME_RUNS>
  <RBI>31</RBI>
  <STEALS>2</STEALS>
  <CAUGHT_STEALING>4</CAUGHT_STEALING>
  <SACRIFICE_HITS>8</SACRIFICE_HITS>
  <SACRIFICE_FLIES>1</SACRIFICE_FLIES>
  <ERRORS>3</ERRORS>
  <WALKS>14</WALKS>
  <STRUCK_OUT>38</STRUCK_OUT>
  <HIT_BY_PITCH>2</HIT_BY_PITCH>
</PLAYER>
```

نلقي نظرة على إحصاءات رامي الكرة وعلى الرغم من أن رماة الكرة في بعض الأحيان في بطولة الدوري الأمريكي يضربون الكرة، وفي أحيان أخرى في بطولة الدوري العام يضربون الكرة عدد مرات أقل من اللاعبين الأخريين فإن إحصاءات رمي الكرة تتضمن عدد المباريات التي لعبت وعدد مرات الفوز والخسارة وعدد جولات رمي الكرة الخ. ما يلي هو إحصاءات عام١٩٩٨ للاعب Hideki Irabu وقد تم كتابتها بأسلوب الرموز في XML:

```
<PLAYER>
  <GIVEN_NAME>Hideki</GIVEN_NAME>
  <SURNAME>Irabu</SURNAME>
  <POSITION>Starting Pitcher</POSITION>
  <WINS>13</WINS>
  <LOSSES>9</LOSSES>
  <SAVES>0</SAVES>
  <GAMES>29</GAMES>
  <GAMES_STARTED>28</GAMES_STARTED>
  <COMPLETE_GAMES>2</COMPLETE_GAMES>
  <SHUT_OUTS>1</SHUT_OUTS>
  <ERA>4.06</ERA>
  <INNINGS>173</INNINGS>
  <HOME_RUNS>148</HOME_RUNS>
  <RUNS>27</RUNS>
  <EARNED_RUNS>79</EARNED_RUNS>s
  <HIT_BATTER>78</HIT_BATTER>
```

```
  <WILD_PITCHES>9</WILD_PITCHES>
  <BALK>6</BALK>
  <WALKED_BATTER>1</WALKED_BATTER>
  <STRUCK_OUT_BATTER>76</STRUCK_OUT_BATTER>
</PLAYER
```

الإيجاز في لغة ترميز XML ليس له أهمية قصوى

تم اتباع مبدأ XML :"الإنجاز في لغة ترميز XML ليس ذو أهمية قصوى" في المثـــال المستخدم لمساعدة أي قارئ لا يعلم الكثير عن رياضة البيسبول وبالتالي لا يعلم الكثير من الاختصارات المعتادة مثل: walk BB (القاعدة على الكرة) وليس W كما قد يتوقـع البعض وإذا كان حجم المستند مهماً فيمكن ضغط الملفات باستخدام تنسيق الملفـات zip أو أي أداة قياسية أخرى.

هذا لا يعني إن مستندات XML تميل إلى الطول وبالتالي تثير الملل أثناء كتابتها بـــاليد وإذا تم استخدام الاختصارات فإن عنصر PLAYER يبدو كما يلي:

```
<PLAYER>
  <GIVEN_NAME>Joe</GIVEN_NAME>
  <SURNAME>Girardi</SURNAME>
  <P>C</P>
  <G>78</G>
  <AB>254</AB>
  <R>31</R>
  <H>70</H>
  <DO>11</DO>
  <TR>4</TR>
  <HR>3</HR>
  <RBI>31</RBI>
  <BB>14</BB>
  <SO>38</SO>
  <SB>2</SB>
  <CS>4</CS>
  <HBP>2</HBP>
</PLAYER>
```

تجميع مستند XML في وحدة واحدة

وصولاً إلى هذا الجزء من الفصل وكان يتم توضيح مستند XML جزء بجزء وعنصر تلو الأخـر والآن يتم وضع كل الأجزاء معاً ولننظر إلى المستند بأكمله وهو يحتـــوي علـــى الإحصـــاءات الخاصة بموسم بطولة الدوري الرئيسي لعام ١٩٩٨. توضح تعليمات برمجة ٤-١ مســتند XML كاملاً وبه بيانات بطولتين للدوري وست مجموعات تنقسم إلى ٣٠ فرقة و ٩ لاعبين.

تعليمات برمجية ٤-١: مستند XML كاملاً

```
<?xml version="1.0"?>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>National League</LEAGUE_NAME>
    <DIVISION>
       <DIVISION_NAME>East</DIVISION_NAME>
         <TEAM>
          <TEAM_CITY>Atlanta</TEAM_CITY>
          <TEAM_NAME>Braves</TEAM_NAME>
          <PLAYER>
           <SURNAME>Malloy</SURNAME>
           <GIVEN_NAME>Marty</GIVEN_NAME>
           <POSITION>Second Base</POSITION>
           <GAMES>11</GAMES>
           <GAMES_STARTED>8</GAMES_STARTED>
           <AT_BATS>28</AT_BATS>
           <RUNS>3</RUNS>
           <HITS>5</HITS>
           <DOUBLES>1</DOUBLES>
           <TRIPLES>0</TRIPLES>
           <HOME_RUNS>1</HOME_RUNS>
           <RBI>1</RBI>
           <STEALS>0</STEALS>
           <CAUGHT_STEALING>0</CAUGHT_STEALING>
```

```
  <SACRIFICE_HITS>0</SACRIFICE_HITS>
  <SACRIFICE_FLIES>0</SACRIFICE_FLIES>
  <ERRORS>0</ERRORS>
  <WALKS>2</WALKS>
  <STRUCK_OUT>2</STRUCK_OUT>
  <HIT_BY_PITCH>0</HIT_BY_PITCH>
 </PLAYER>
 <PLAYER>
  <SURNAME>Guillen</SURNAME>
  <GIVEN_NAME>Ozzie </GIVEN_NAME>
  <POSITION>Shortstop</POSITION>
  <GAMES>83</GAMES>
  <GAMES_STARTED>59</GAMES_STARTED>
  <AT_BATS>264</AT_BATS>
  <RUNS>35</RUNS>
  <HITS>73</HITS>
  <DOUBLES>15</DOUBLES>
  <TRIPLES>1</TRIPLES>
  <HOME_RUNS>1</HOME_RUNS>
  <RBI>22</RBI>
  <STEALS>1</STEALS>
  <CAUGHT_STEALING>4</CAUGHT_STEALING>
  <SACRIFICE_HITS>4</SACRIFICE_HITS>
  <SACRIFICE_FLIES>2</SACRIFICE_FLIES>
  <ERRORS>6</ERRORS>
  <WALKS>24</WALKS>
  <STRUCK_OUT>25</STRUCK_OUT>
  <HIT_BY_PITCH>1</HIT_BY_PITCH>
</PLAYER>
<PLAYER>
<SURNAME>Bautista</SURNAME>
  <GIVEN_NAME>Danny</GIVEN_NAME>
  <POSITION>Outfield</POSITION>
  <GAMES>82</GAMES>
```

```
  <GAMES_STARTED>27</GAMES_STARTED>
  <AT_BATS>144</AT_BATS>
  <RUNS>17</RUNS>
  <HITS>36</HITS>
  <DOUBLES>11</DOUBLES>
  <TRIPLES>0</TRIPLES>
  <HOME_RUNS>3</HOME_RUNS>
  <RBI>17</RBI>
  <STEALS>1</STEALS>
  <CAUGHT_STEALING>0</CAUGHT_STEALING>
  <SACRIFICE_HITS>3</SACRIFICE_HITS>
  <SACRIFICE_FLIES>2</SACRIFICE_FLIES>
  <ERRORS>2</ERRORS>
  <WALKS>7</WALKS>
  <STRUCK_OUT>21</STRUCK_OUT>
  <HIT_BY_PITCH>0</HIT_BY_PITCH>
</PLAYER>
<PLAYER>
  <SURNAME>Williams</SURNAME>
  <GIVEN_NAME>Gerald</GIVEN_NAME>
  <POSITION>Outfield</POSITION>
  <GAMES>129</GAMES>
  <GAMES_STARTED>51</GAMES_STARTED>
  <AT_BATS>266</AT_BATS>
  <RUNS>46</RUNS>
  <HITS>81</HITS>
  <DOUBLES>18</DOUBLES>
  <TRIPLES>3</TRIPLES>
  <HOME_RUNS>10</HOME_RUNS>
  <RBI>44</RBI>
  <STEALS>11</STEALS>
  <CAUGHT_STEALING>5</CAUGHT_STEALING>
  <SACRIFICE_HITS>2</SACRIFICE_HITS>
  <SACRIFICE_FLIES>1</SACRIFICE_FLIES>
```

```
<ERRORS>5</ERRORS>
      <WALKS>17</WALKS>
      <STRUCK_OUT>48</STRUCK_OUT>
      <HIT_BY_PITCH>3</HIT_BY_PITCH>
    </PLAYER>
    <PLAYER>
      <SURNAME>Glavine</SURNAME>
      <GIVEN_NAME>Tom</GIVEN_NAME>
      <POSITION>Starting Pitcher</POSITION>
      <WINS>20</WINS>
      <LOSSES>6</LOSSES>
      <SAVES>0</SAVES>
      <GAMES>33</GAMES>
      <GAMES_STARTED>33</GAMES_STARTED>
      <COMPLETE_GAMES>4</COMPLETE_GAMES>
      <SHUT_OUTS>3</SHUT_OUTS>
      <ERA>2.47</ERA>
      <INNINGS>229.1</INNINGS>
      <HOME_RUNS>202</HOME_RUNS>
      <RUNS>13</RUNS>
      <EARNED_RUNS>67</EARNED_RUNS>
      <HIT_BATTER>63</HIT_BATTER>
      <WILD_PITCHES>2</WILD_PITCHES>
      <BALK>3</BALK>
      <WALKED_BATTER>0</WALKED_BATTER>
      <STRUCK_OUT_BATTER>74</STRUCK_OUT_BATTER>
    </PLAYER>
    <PLAYER>
      <SURNAME>Lopez</SURNAME>
      <GIVEN_NAME>Javier</GIVEN_NAME>
      <POSITION>Catcher</POSITION>
      <GAMES>133</GAMES>
      <GAMES_STARTED>124</GAMES_STARTED>
      <AT_BATS>489</AT_BATS>
```

```
  <RUNS>73</RUNS>
  <HITS>139</HITS>
  <DOUBLES>21</DOUBLES>
  <TRIPLES>1</TRIPLES>
  <HOME_RUNS>34</HOME_RUNS>
  <RBI>106</RBI>
  <STEALS>5</STEALS>
  <CAUGHT_STEALING>3</CAUGHT_STEALING>
  <SACRIFICE_HITS>1</SACRIFICE_HITS>
  <SACRIFICE_FLIES>8</SACRIFICE_FLIES>
  <ERRORS>5</ERRORS>
  <WALKS>30</WALKS>
  <STRUCK_OUT>85</STRUCK_OUT>
  <HIT_BY_PITCH>6</HIT_BY_PITCH></PLAYER>
<PLAYER>
  <SURNAME>Klesko</SURNAME>
  <GIVEN_NAME>Ryan</GIVEN_NAME>
  <POSITION>Outfield</POSITION>
  <GAMES>129</GAMES>
  <GAMES_STARTED>124</GAMES_STARTED>
  <AT_BATS>427</AT_BATS>
  <RUNS>69</RUNS>
  <HITS>117</HITS>
  <DOUBLES>29</DOUBLES>
  <TRIPLES>1</TRIPLES>
  <HOME_RUNS>18</HOME_RUNS>
  <RBI>70</RBI>
  <STEALS>5</STEALS>
  <CAUGHT_STEALING>3</CAUGHT_STEALING>
  <SACRIFICE_HITS>0</SACRIFICE_HITS>
  <SACRIFICE_FLIES>4</SACRIFICE_FLIES>
  <ERRORS>2</ERRORS>
  <WALKS>56</WALKS>
  <STRUCK_OUT>66</STRUCK_OUT>
```

```
<HIT_BY_PITCH>3</HIT_BY_PITCH></PLAYER>
<PLAYER>
  <SURNAME>Galarraga</SURNAME>
  <GIVEN_NAME>Andres</GIVEN_NAME>
  <POSITION>First Base</POSITION>
  <GAMES>153</GAMES>
  <GAMES_STARTED>151</GAMES_STARTED>
  <AT_BATS>555</AT_BATS>
  <RUNS>103</RUNS>
  <HITS>169</HITS>
  <DOUBLES>27</DOUBLES>
  <TRIPLES>1</TRIPLES>
  <HOME_RUNS>44</HOME_RUNS>
  <RBI>121</RBI>
  <STEALS>7</STEALS>
  <CAUGHT_STEALING>6</CAUGHT_STEALING>
  <SACRIFICE_HITS>0</SACRIFICE_HITS>
  <SACRIFICE_FLIES>5</SACRIFICE_FLIES>
  <ERRORS>11</ERRORS>
  <WALKS>63</WALKS>
  <STRUCK_OUT>146</STRUCK_OUT>
  <HIT_BY_PITCH>25</HIT_BY_PITCH></PLAYER>
<PLAYER>
  <SURNAME>Helms</SURNAME>
  <GIVEN_NAME>Wes</GIVEN_NAME>
  <POSITION>Third Base</POSITION>
  <GAMES>7</GAMES>
  <GAMES_STARTED>2</GAMES_STARTED>
  <AT_BATS>13</AT_BATS>
  <RUNS>2</RUNS>
  <HITS>4</HITS>
  <DOUBLES>1</DOUBLES>
  <TRIPLES>0</TRIPLES>
  <HOME_RUNS>1</HOME_RUNS>
```

```
        <RBI>2</RBI>
  <STEALS>0</STEALS>
        <CAUGHT_STEALING>0</CAUGHT_STEALING>
        <SACRIFICE_HITS>0</SACRIFICE_HITS>
        <SACRIFICE_FLIES>0</SACRIFICE_FLIES>
        <ERRORS>1</ERRORS>
        <WALKS>0</WALKS>
        <STRUCK_OUT>4</STRUCK_OUT>
        <HIT_BY_PITCH>0</HIT_BY_PITCH></PLAYER>
      </TEAM>
      <TEAM>
       <TEAM_CITY>Florida</TEAM_CITY>
       <TEAM_NAME>Marlins</TEAM_NAME>
      </TEAM>
      <TEAM>
       <TEAM_CITY>Montreal</TEAM_CITY>
       <TEAM_NAME>Expos</TEAM_NAME>
      </TEAM>
      <TEAM>
       <TEAM_CITY>New York</TEAM_CITY>
       <TEAM_NAME>Mets</TEAM_NAME>
      </TEAM>
      <TEAM>
       <TEAM_CITY>Philadelphia</TEAM_CITY>
      <TEAM_NAME>Phillies</TEAM_NAME>
      </TEAM>
    </DIVISION>
    <DIVISION>
      <DIVISION_NAME>Central</DIVISION_NAME>
      <TEAM>
       <TEAM_CITY>Chicago</TEAM_CITY>
       <TEAM_NAME>Cubs</TEAM_NAME>
      </TEAM>
      <TEAM>
```

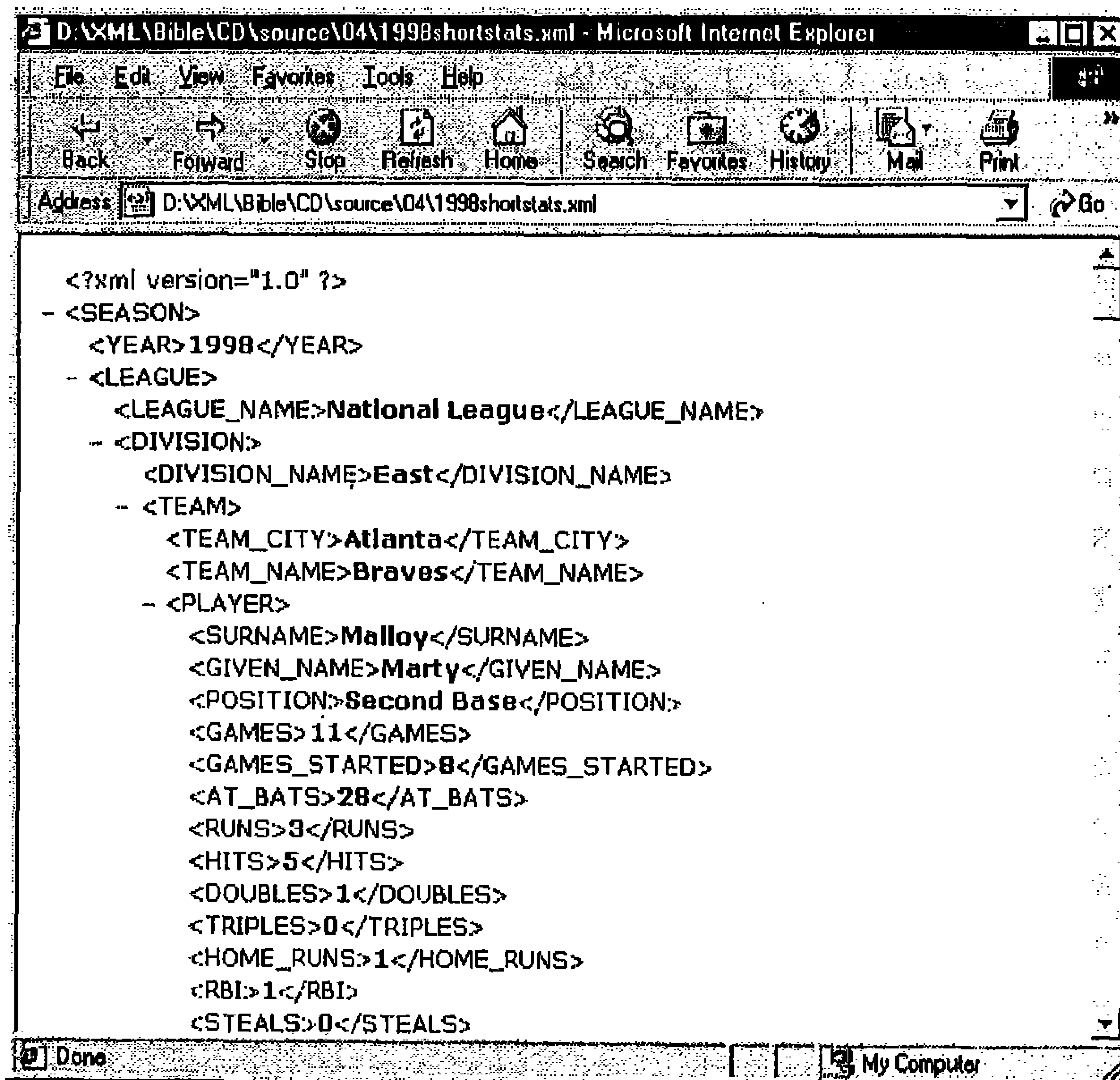
```xml
      <TEAM_CITY>Cincinatti</TEAM_CITY>
      <TEAM_NAME>Reds</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Houston</TEAM_CITY>
      <TEAM_NAME>Astros</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Milwaukee</TEAM_CITY>
      <TEAM_NAME>Brewers</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Pittsburgh</TEAM_CITY>
      <TEAM_NAME>Pirates</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>St. Louis</TEAM_CITY>
      <TEAM_NAME>Cardinals</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Arizona</TEAM_CITY>
      <TEAM_NAME>Diamondbacks</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Colorado</TEAM_CITY>
      <TEAM_NAME>Rockies</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Los Angeles</TEAM_CITY>
      <TEAM_NAME>Dodgers</TEAM_NAME>
    </TEAM>
```

```xml
    <TEAM>
      <TEAM_CITY>San Diego</TEAM_CITY>
      <TEAM_NAME>Padres</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>San Francisco</TEAM_CITY>
      <TEAM_NAME>Giants</TEAM_NAME>
    </TEAM>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American League</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Baltimore</TEAM_CITY>
      <TEAM_NAME>Orioles</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Boston</TEAM_CITY>
      <TEAM_NAME>Red Sox</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>New York</TEAM_CITY>
      <TEAM_NAME>Yankees</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Tampa Bay</TEAM_CITY>
      <TEAM_NAME>Devil Rays</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Toronto</TEAM_CITY>
      <TEAM_NAME>Blue Jays</TEAM_NAME>
    </TEAM>
```

```
    </DIVISION>
    <DIVISION>
  <DIVISION_NAME>Central</DIVISION_NAME>
      <TEAM>
        <TEAM_CITY>Chicago</TEAM_CITY>
        <TEAM_NAME>White Sox</TEAM_NAME>
      </TEAM>
      <TEAM>
        <TEAM_CITY>Kansas City</TEAM_CITY>
        <TEAM_NAME>Royals</TEAM_NAME>
      </TEAM>
      <TEAM>
        <TEAM_CITY>Detroit</TEAM_CITY>
        <TEAM_NAME>Tigers</TEAM_NAME>
      </TEAM>
      <TEAM>
        <TEAM_CITY>Cleveland</TEAM_CITY>
        <TEAM_NAME>Indians</TEAM_NAME>
      </TEAM>
      <TEAM>
        <TEAM_CITY>Minnesota</TEAM_CITY>
        <TEAM_NAME>Twins</TEAM_NAME>
      </TEAM>
    </DIVISION>
    <DIVISION>
      <DIVISION_NAME>West</DIVISION_NAME>
       <TEAM>
         <TEAM_CITY>Anaheim</TEAM_CITY>
         <TEAM_NAME>Angels</TEAM_NAME>
       </TEAM>
       <TEAM>
         <TEAM_CITY>Oakland</TEAM_CITY>
         <TEAM_NAME>Athletics</TEAM_NAME>
       </TEAM>
```

```
    <TEAM>
      <TEAM_CITY>Seattle</TEAM_CITY>
      <TEAM_NAME>Mariners</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Texas</TEAM_CITY>
      <TEAM_NAME>Rangers</TEAM_NAME>
    </TEAM>
  </DIVISION>
 </LEAGUE>
</SEASON>
```

يوضح الشكل ٤-١ هذا المستند وقد تم تحميلة إلى Internet Explorer 5.0.



الشكل ٤-١ عرض إحصاءات بطولة الـدوري الرئيسـي لعـام ١٩٩٨ فـي Internet Explorer 5.0.

مازال هذا المستند يعتبر غير كاملاً لأنه يحتوي على ٩ لاعبين من فريق واحد فقـط وهـو فريق Atlanta Braves وتوضيح أكثر من هذا يجعل المثال أكبر من إن يتم تضمينه في الكتاب.

مزايا تنسيق XML

يعرض جدول ٤-١ بيانات ضرب الكرة لفريق بطريقة مضغوطة ومفهومة ولكن يقدم مثال ٤-١ نفس البيانات بعد إعادة كتابتها بأسلوب مطول في مستند XML ولهذا الأسلوب مزاياه ومنها:

- شرح البيانات لنفسها.
- يمكن التعامل مع البيانات بواسطة الأدوات القياسية.
- يتم عرض البيانات بواسطة الأدوات القياسية.
- يسهل إنشاء طرق عرض مختلفة لنفس البيانات باستخدام أوراق النمط.

الفائدة الأساسية لاستخدام تنسيق XML هو قيام البيانات بتوضيح نفسها حيث يرتبط معني كل رقم بالرقم نفسه وبطريقة لا يمكن إساءة فهمها فمثلا عند قراءة المستند تعلم إن ١٢١ الموجود في <HITS>121</HITS> تشير إلى الأهداف وليس إلى batted in runs أو strikeouts وإذا اسقط من يقوم بكتابة هذا المستند رقم فإن هذا لا يعني إن أي رقم يتبعه لن يتم تفسيره بطريقة صحيحة فمثلا HITS تظل HITS حتى وان كان عنصر RUNS مفقود.



تجد في الجزء الثاني أن XML يستخدم DTDs لفرض قيود لظهور عناصر مثل HITS أو RUNS.

الفائدة الثانية من تقديم البيانات في مستند XML هو أن مستند XML يتيح التعامل مع البيانات بأدوات تتيحها XML وتأتي تلك الأدوات من شركات برامج مكلفة مثل Adobe FrameMaker وكذلك من مصادر برنامج مفتوحة مجاناً للمستخدمين مثل Python و perl. قد تكون البيانات كبيرة الحجم ولكن يتيح الإسهاب استخدام المزيد من الأدوات لمعالجتها.

يصبح نفس الشيء صحيح عند البدء في عرض البيانات حيث يمكن تحميل مستند XML إلى Internet Explorer 5.0 أو Mozilla أو FrameMaker5.5.6 والعديد من الأدوات الأخرى حيث يقدموا عرض متميز ومفيد للبيانات ويمكن تحميل البيانات إلى محرر نص أساسي وبسيط مثل BBEdit أو vi أو TextPad وبهذا تكون مرئية ولو بصورة هامشية على معظم أنظمة التشغيل.

استخدام برامج جديدة ليس هو الطريقة الوحيدة للحصول على عرض مختلف للبيانات ففي الجزء التالي يتم بناء ورقة نمط لإحصاءات رياضية البيسبول والتي تقدم طريقه مختلفة تماماً لرؤية البيانات عما ظهر في الشكل ٤-١ وفي كل مرة يتم تطبيق ورقة مختلفة إلى نفس المستند تظهر صورة مختلفة وجديدة.

في النهاية إذا كان حجم المستند مهم فإن محركات الأقراص الجديدة كبيرة جـــداً وتســتطيع تحميل كمية ضخمة من البيانات حتى وان لم يتم تخزينها بصورة جيدة، بالإضافة إلى ذلك يمكــن ضغط ملفات XML بصورة جيدة ويزن مستند إحصاءات بطولة الدوري الرئيســي لعــام ١٩٩٨ كاملة ٦٥٣ كيلو ومع ذلك يمكن ضغط الملف باستخدام تنسيق الملف gzip ليصل حجمه إلى ٦٦ كيلو وهو بذلك يقل نسبة ٩٠% تقريباً. تستطيع بعض أنواع الخادم المتقدم HTTP مثــل Jigsaw إرسال ملفات مضغوطة بدلاً من الملفات غير المضغوطة وبذلك فأن حــــيز الشــبكة المســتخدم بواسطة مستند بمثل هذا الحجم يماثل تقريباً محتوى المعلومات الفعلي للمستند. ولا يجب افــترض إن تنسيقات الملف الثنائية وبخاصة ذات الأهداف العامة هي بالضرورة أكـــثر فاعليـــة.فملـــف مايكروسوفت اكسل الذي يحتوي على نفس البيانات مثــــل ملـــف 1998statistics.xml يـــأخذ بالتحديد ٢٠٣٧ ميجابايت وهي مساحة أكثر ثلاث مرات من المساحة السابقة وعلى الرغم مـــن إمكانية إنشاء تنسيقات ملفات أكثر كفاءة لهذه البيانات وأيضاً تشفيرها وترميزها بصـــورة أكــثر فاعلية ففي التطبيق العملي لا يعتبر هذا ضروريا.

إعداد ورقة نمط لعرض مستند

عرض مستند XML الأساسي الموضح في الشكل "٤-١" له عدة استخدامات، على سبيل المثـــال يتيح طي وتوسيع عناصر فردية ليمكن رؤية الأجزاء المراد رؤيتها من المستند فقط وليتم رؤيـــة المستند بشكل واضح وكامل وللحصول على صورة كاملة منه لعرضها على الويب يجب كتابـــة ورقة نمط للمستند.

نستخدم في هذا الفصل أوراق نمط CSS وتربط أوراق نمط CSS بين تنسيقات محددة لكـــل عنصر في المستند والقائمة الكاملة العناصر المستخدمة في مستند XML هي:

SEASON
YEAR
LEAGUE
LEAGUE_NAME
DIVISION
DIVISION_NAME
TEAM
TEAM_CITY
TEAM_NAME
PLAYER
SURNAME

GIVEN_NAME
POSITION
GAMES
GAMES_STARTED
AT_BATS
RUNS
HITS
DOUBLES
TRIPLES
HOME_RUNS
RBI
STEALS
CAUGHT_STEALING
SACRIFICE_HITS
SACRIFICE_FLIES
ERRORS
WALKS
STRUCK_OUT
HIT_BY_PITCH

عامة، يجب اتباع إجراء تكراري بإضافة قواعد نمط لكل من هذه العناصر الواحد تلو الآخر والتحقق من أنهم يؤدون ما ترغبه ثم التحرك للعنصر الذي يلي ذلك. وفي هذا المثال يسهم استخدام مثل هذا الأسلوب في تعريف مزايا خصائص CSS الواحدة تلو الأخرى لمن لا يعرفها من المستخدمين.

الارتباط إلى ورقة نمط

يمكن تسمية ورقة النمط أي اسم فإذا كانت تطبق على مستند واحد فقط فمن المعتاد إعطائها نفس اسم المستند ولكن يكون ملحق الاسم هو css بدلاً من xml. على سبيل المثال ورقة نمط مستند XML المسمي 1998shortstats.xml يمكن أن تسمي 1998shortstats.css. من جهة أخرى إذا تم تطبيق نفس ورقة النمط على عدة مستندات فيجب إعطائها اسم أكثر عمومية مثل baseballstats.css.

المرجع يمكن تطبيق أكثر من ورقة نمط على نفس المستند وذلك منذ ظهور أوراق نمط CSS ولهذا فمن الممكن تطبيق ورقة نمط baseballstats CSS على بعض قواعد التنسيق العامة بينما تتجاوز ورقة نمط 1998shortstats.css عن بعض الأشياء لتتعامل مع بيانات محددة في مستند 1998shortstats.xml وتتم مناقشة هذا الإجراء في الفصل ١٢ "أوراق نمط المستوى ١".

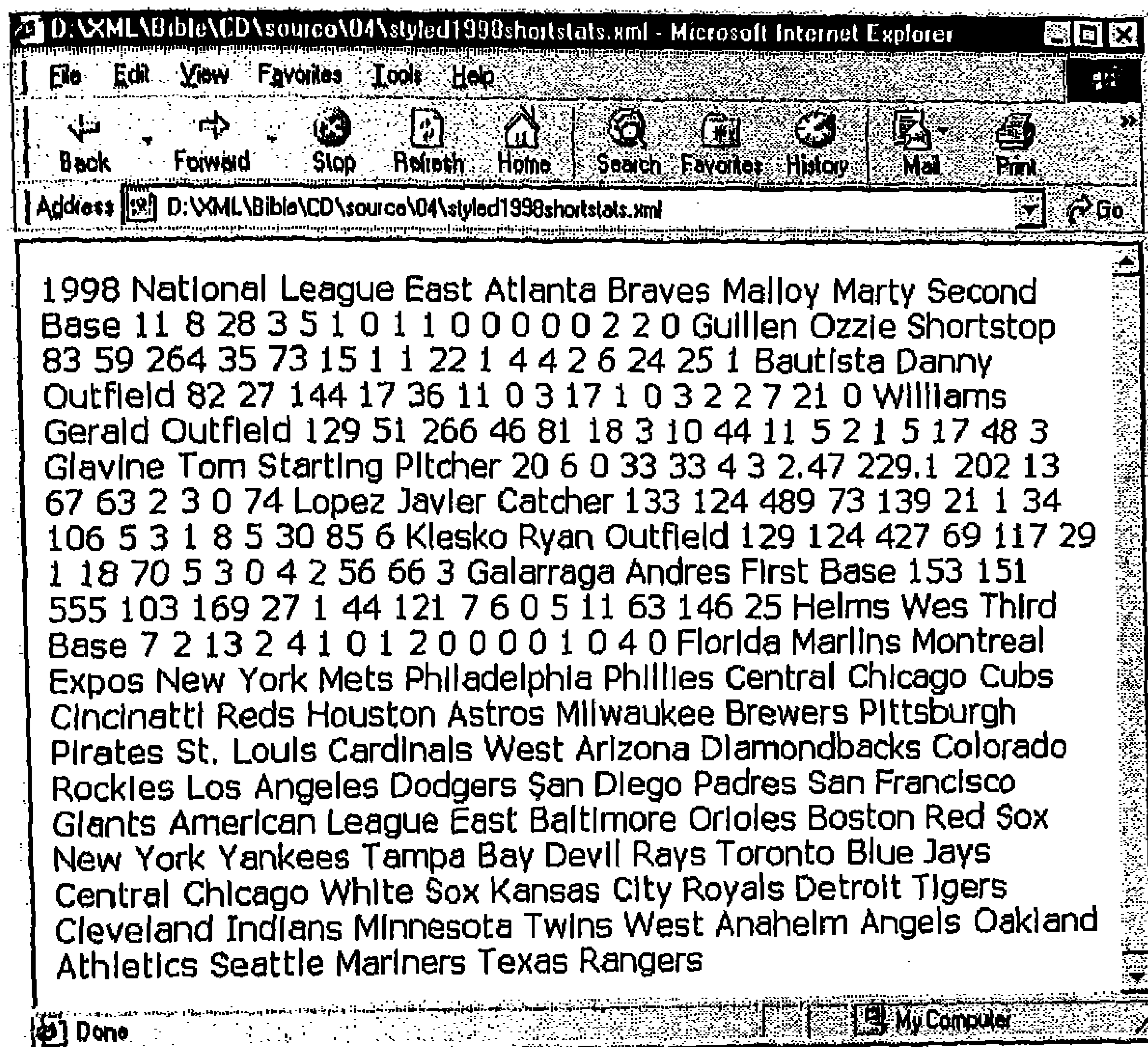
لإلحاق ورقة نمط إلى المستند يجب إضافة تعليمات المعالجة <?xml-stylesheet?> بين تعريف XML والعنصر الجذري مثل ما يلي

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="baseballstats.css"?>
<SEASON>
```

ينبه هذا الإجراء المستعرض الذي يقوم بقراءة المستند بتطبيق ورقة النمط الموجودة في ملف baseballstats.css على هذا المستند ومن المفترض أن يظل هذا الملف في نفس الدليل وعلى نفس الخادم مثل مستند XML وهذا يعني إن ملف baseballstats.css هو برنامج URL مناسب ويمكن كذلك استخدام URLs كاملة ومثال على لذلك:

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css"
href="http://metalab.unc.edu/xml/examples/baseballstats.css"?>
<SEASON>
```

يمكن البدء يوضح ملف فارغ يسمي baseballstats.css في نفس دليل مستند XML. بمجرد الانتهاء من ذلك وإضافة تعليمات معالجة إلى 1998shortstats.xml "تعليمات البرمجة ٤-١" يظهر المستند كما يوضح الشكل "٤-٢" ويظهر فقط محتوى العنصر وبهذا يختفي العرض المفصل الذي يمكن طيه لشكل "٤-١". يستخدم تنسيق محتوى العنصر القيم الافتراضية للمستعرض وهي في هذه الحالة الخط الأسود Times Roman وإعداده على ١٢ نقطة على خلفية بيضاء.



الشكل ٤-٢ عرض إحصاءات بطولة الدوري الرئيسي لعام ١٩٩٨ بعد تطبيق ورقة نمط فارغة.



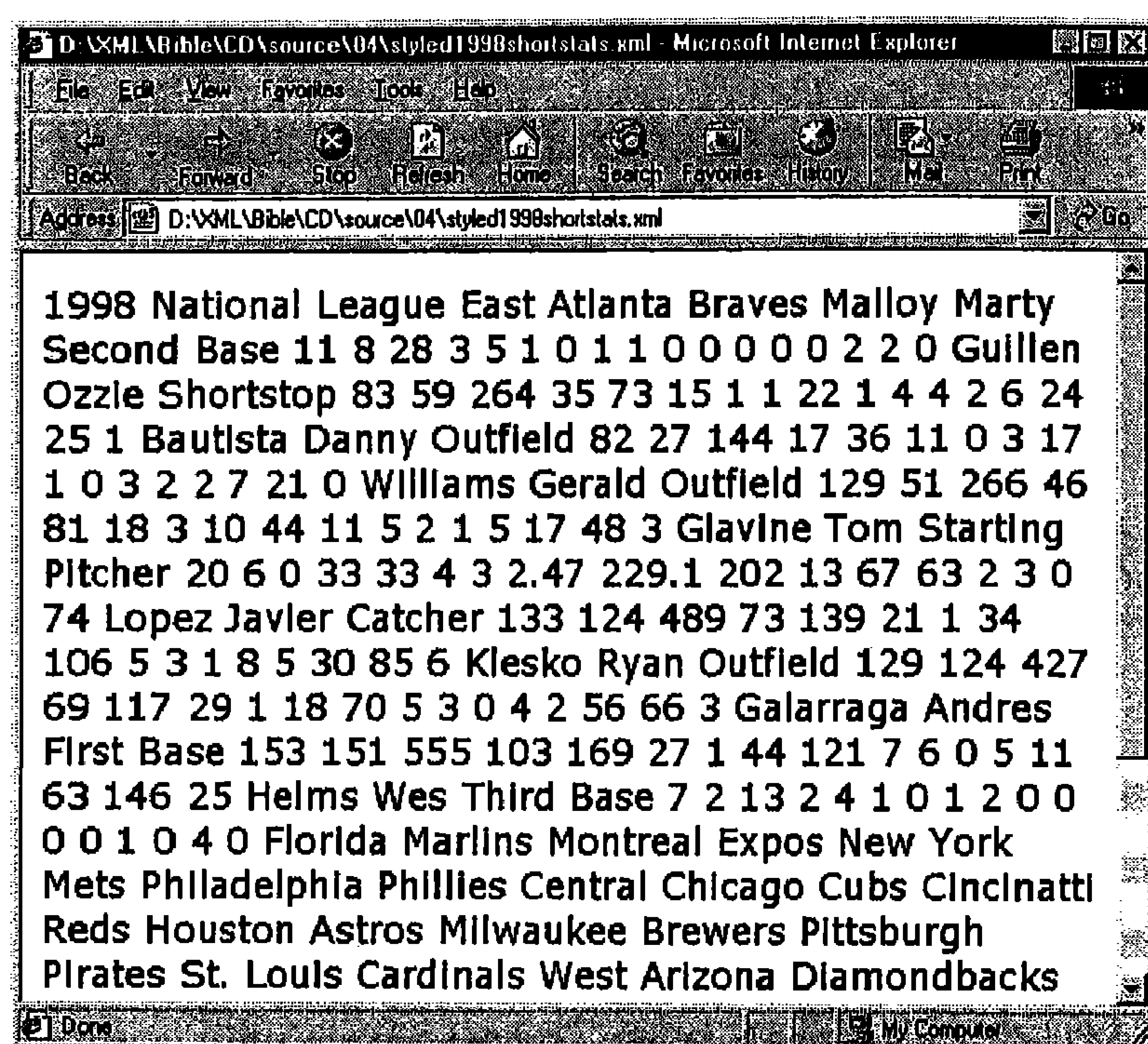
إذا لم توجد تعليمات معالجة ورقة النمط المسماة xml-stylesheet في الموقع المحدد سيظهر شكل يقترب كثيراً من الشكل الموضح في "٤-٢".

تعيين قواعد النمط إلى العنصر الجذري

لا يجب تعيين قاعدة نمط لكل عنصر في القائمة حيث تسمح العديد من العناصر لأنماطها الأصلية إن تتالي عليها ولهذا فإن أهم نمط هو النمط الذي يتم تعيينه للعنصر الجذري وفي هذا المثال SEASON هو العنصر الجذري ويقوم هذا التعريف بتحديد الافتراضات لكل العناصر الأخرى الموجودة على الصفحة. لا تكون دقة شاشات أجهزة الكمبيوتر وهي معدة على ٧٢ نقطة في البوصة تقريباً بمثل دقة الورق وهو معد على ٣٠٠ نقطة في البوصة أو أكثر ولهذا السبب يجب أن تستخدم صفحات الويب حجم نقاط اكبر من المعتاد وليكون الإعداد الافتراضي الذي نستخدمه نوعه ١٤ نقطة أسود على خلفية بيضاء كما هو موضح اسفل الكلام.

```
SEASON {font-size: 14pt; background-color: white;
      color: black; display: block}
```

ضع هذه العبارة في ملف نص وأحفظ الملف باسم baseballstats css فـــي نفـــس الدليـــل الموضوع فيه تعليمات برمجية "٤-١" 1998shortstats.xml وافتـــح 1998shortstats.xml في المستعرض وما يظهر هو شكل الموضح في الشكل "٤-٣".



الشكل ٤-٣ إحصاءات البيسبول مكتوبة بحجم خط ١٤ نقطة أسود ابيض

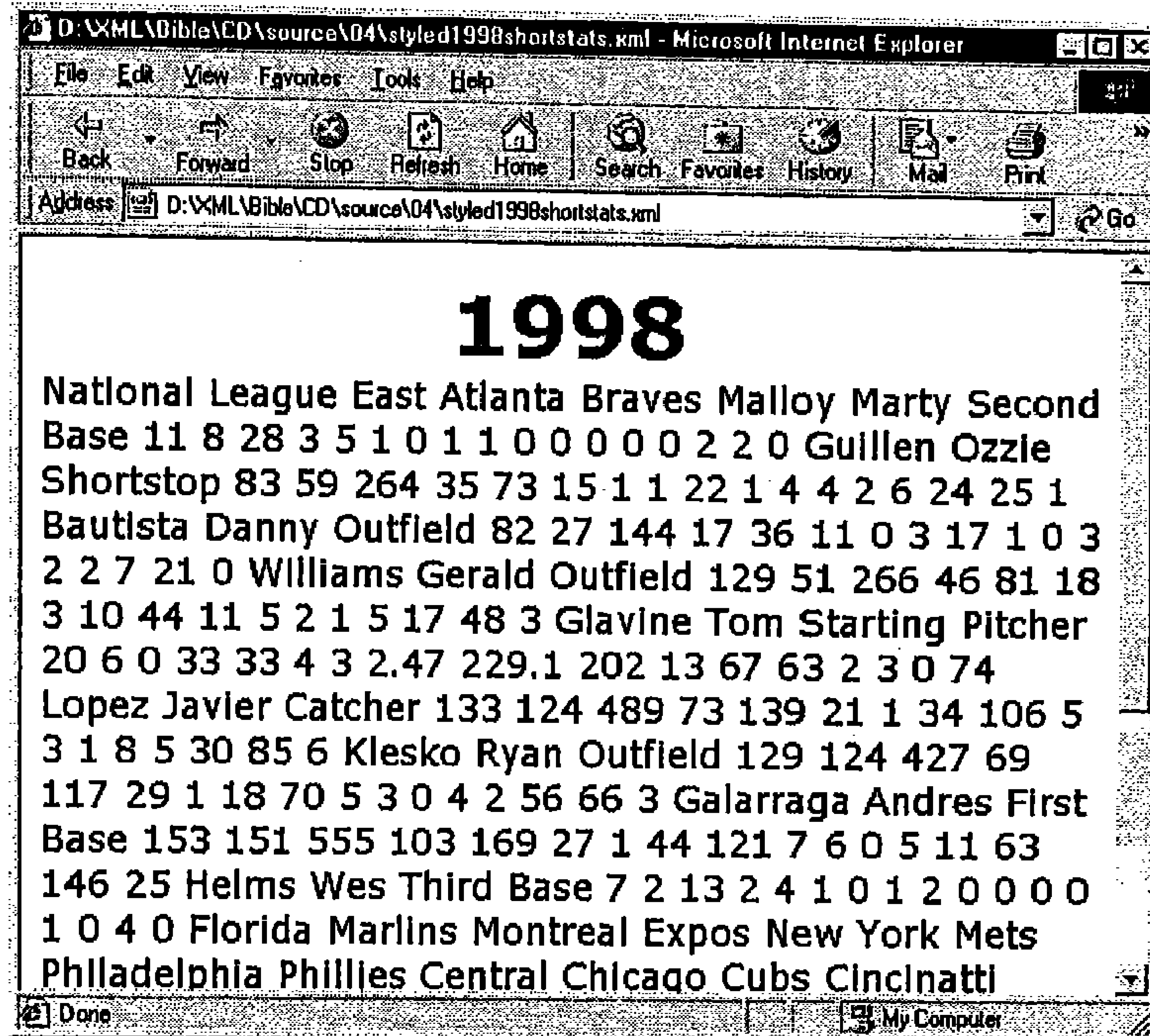
تغير حجم الخط الافتراضي بين الشكل "٤-٢" والشكل "٤-٣" ولكن لم يتغير لون النـــص أو الخلفية وبالطبع لم تكن هناك حاجة ضرورية لأعدادهم لان الخلفية البيضاء والمقدمة السوداء هو الإعداد الافتراضي ولكن إعدادهم لا يضر في شيء.

تعيين قواعد نمط للعناوين

يعتبر عنصر YEAR هو عنوان المستند ولذلك سنجعله كبير وسميك ويتم إعداده إلى حجـــم ٢٣ نقطة وأيضاً يجب إن يكون بعيداً عن محتوى المستند ولهذا سنجعله عنصر متوسط أي يتوســـط صفحة المستند والقاعدة التي ستنفذ لنا ذلك هي:

```
YEAR {display: block; font-size: 32pt; font-weight: bold;
    text-align: center}
```

يوضح الشكل "٤-٤" المستند بعد تطبيق هذه القاعدة علي ورقة النمط ولاحظ فاصل السطر بعد ١٩٩٨ وقد ظهر هذا الفاصل لان عنصر YEAR أصبح الآن عنصر مستوى كتلة -block level element أما بالنسبة لبقية المستندات فهي عناصر محولة ويمكن فقط توسيطها أو محاذاتها لليمين أو اليسار أو ضبط عناصر مستوى الكتلة block-level element.



الشكل ٤-٤ وضع عنصر YEAR في نمط العنوان

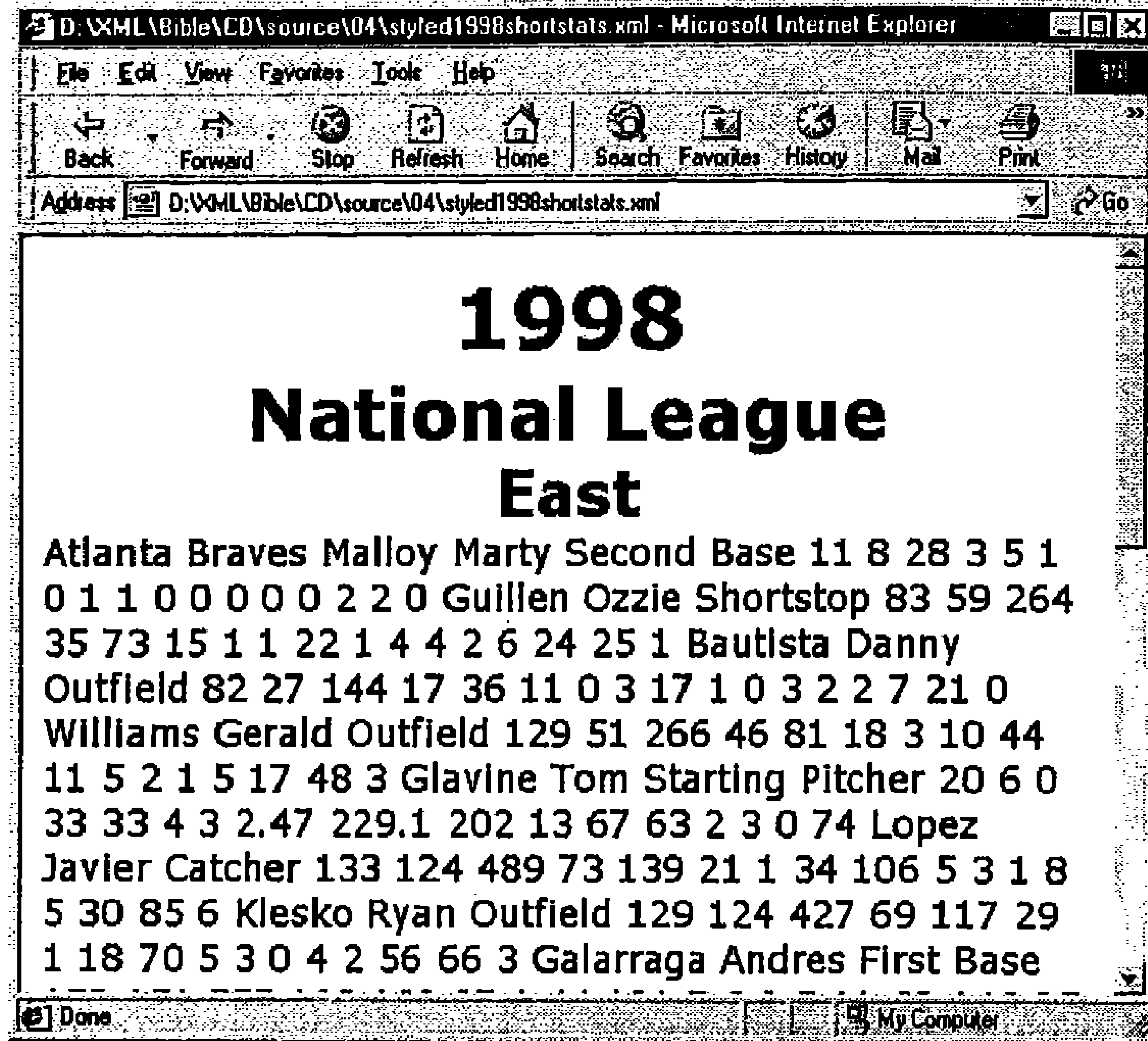
يجعل استخدام قاعدة النمط هذه في هذا المستند عنصر YEAR يكرر وظائف عنصر رأس صفحة HTML H1 وحيث أن هذا المستند هيكلي فإن عناصر أخرى تلعب دور رؤوس الصفحات H2 ورؤوس الصفحات H3 ...الخ ويمكن تنسيق هذه العناصر بقواعد مماثلة ولكن بحجم خط اصغر نسبياً.

فمثلاً ينقسم عنصر SEASON إلى عنصرين LEAGUE ولكل اسم عنصر LEAGUE وهو عنصر LEAGUE_NAME نفس دور عنصر H2 في HTML وبعد ذلك ينقسم كل عنصر LEAGUE إلى ثلاثة عناصر DIVISION واسم كل DIVISION وهو عنصر DIVISION_NAME له نفس دور عنصر H3 في HTML وتقوم القاعدتين التاليتين تنسيقهم:

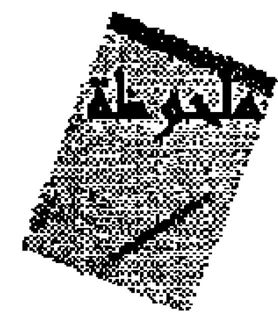
```
LEAGUE_NAME {display: block; text-align: center; font-size: 28pt; font-
weight: bold}
```

DIVISION_NAME {display: block; text-align: center; font-size: 24pt; font-weight: bold}

يوضح الشكل "٤-٥" المستند الذي يظهر بعد تطبيق هذه القاعدة.



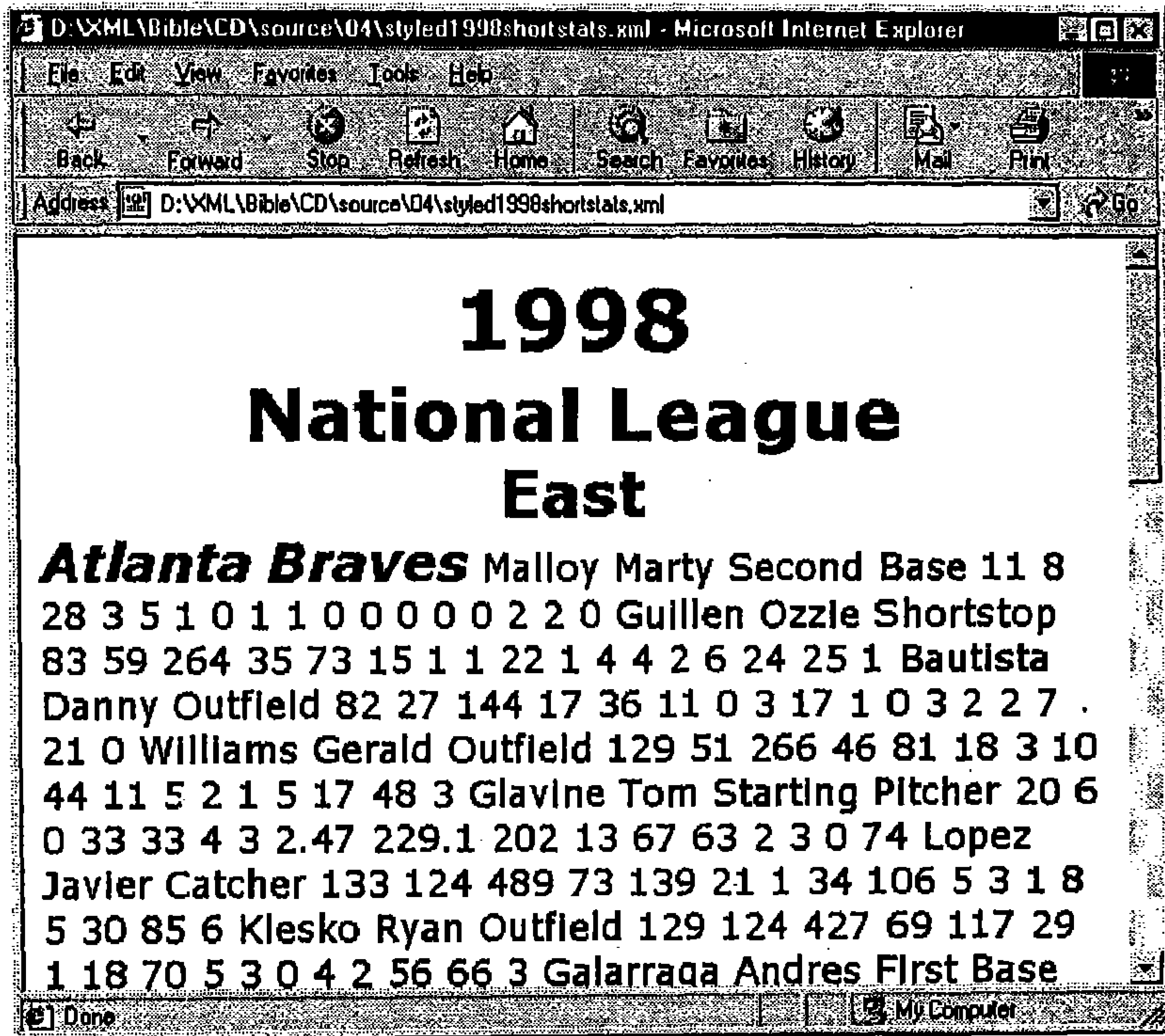
الشكل ٤-٥ وضع نمط عناصر LEAGUE_NAME DIVISION_NAME في رأس الصفحة.



إحدى الفروق الرئيسية بين HTML وXML هو انه في HTML لا يوجد عنصر واحد يحتوي على كلا من عنوان المقطع "رأس الصفحة H2, H3, H4" والمحتوى الكامل للمقطع ويجب بدلاً من ذلك تضمن محتويات المقطع وهي كل شيء بين نهاية مستوى واحد من رأس الصفحة وبداية رأس الصفحة التالي على نفس المستوى ويعتبر هذا الشيء هام جداً بالنسبة للبرامج التي تقوم بتوزيع مستندات HTML لتقوم مثلاً بإنشاء جدول محتويات آليا.

تنقسم المجموعات إلى عناصر TEAM ويتطلب تنسيق هنا العناصر بعض المهارة حيث إن عنوان الفريق ليس فقط عنصر TEAM_NAME ولكن عنصر TEAM_CITY في تسلسل مع TEAM_NAME ولذلك يجب أن تكون تلك العناصر عناصر محولة بدلا من عناصر مستوى الكتلة منفصلة.وعلى الرغم من ذلك فهم مازالوا عناوين ولذلك تم إعدادهم على نوع سميك ومائل وحجم خط ٢٠ نقطة ويوضح الشكل "٤-٦" نتائج إضافة هاتين القاعدتين لورقة النمط.

```
<TEAM>
 <TEAM_TITLE>
  <TEAM_CITY>Colorado</TEAM_CITY>
  <TEAM_NAME>Rockies</TEAM_NAME>
 </TEAM_TITLE>
</TEAM>
```



الشكل ٤-٦ وضع أسماء الفرق في نمط.

يكون من المناسب في هذه المرحلة ترتيب أسماء الفرق والمدن كعنصر مستوى كتلة موحد وهناك عدة طرق لفعل ذلك فمثلا يمكن إضافة عنصر TEAM_TITLE إضافي لمستند XML الذي يكون هدفه الأوحد هو احتواء عنصري TEAM_NAME و TEAM_CITY.ومثال على لذلك

```
<TEAM>
 <TEAM_TITLE>
  <TEAM_CITY>Colorado</TEAM_CITY>
  <TEAM_NAME>Rockies</TEAM_NAME>
 </TEAM_TITLE>
</TEAM>
```

يلي ذلك إضافة قاعدة نمط تطبق تنسيق مستوى الكتلة على TEAM-TITLE وهي:

:TEAM_TITLE {display: block; text-align: center}

لا يجب إعادة تنظيم مستند XML فقط لتجعل ورقة النمط تعمـــل بطريقـــة أســـهل فالـــهدف الأساسي من أوراق النمط هو الاحتفاظ بمعلومات التنسيق خارج المستند نفسه. مع ذلـــك يمكـــن الحصول على نفس التأثير بجعل العناصر التي تسبق وتتبع TEAM وPLAYER عناصر مستوى كتلة. هذا الإجراء يضع عنصر TEAM NAME وعنصر TEAM CLTY في عنصر مستوى كتلة ضمني خاص بهم ويوضح الشكل (٤-٧) نتائج ذلك:

```
TEAM {display: block}
PLAYER {display: block}
```

D:\XML\Bible\CD\source\04\styled1998shortstats.xml - Microsoft Internet Explorer

1998
National League
East
Atlanta Braves
Malloy Marty Second Base 11 8 28 3 5 1 0 1 1 0 0 0 0 0 2 2 0
Guillen Ozzie Shortstop 83 59 264 35 73 15 1 1 22 1 4 4 2 6 24 25 1
Bautista Danny Outfield 82 27 144 17 36 11 0 3 17 1 0 3 2 2 7 21 0
Williams Gerald Outfield 129 51 266 46 81 18 3 10 44 11 5 2 1 5 17 48 3
Glavine Tom Starting Pitcher 20 6 0 33 33 4 3 2.47

الشكل ٤-٧ وضع أسماء الفرق والمدن في نمط رؤوس الصفحة.

تعيين قواعد النمط لعناصر الإحصاءات واللاعبين

من أكثر التنسيقات التي تتطلب مهارة هي تنسيق اللاعبين الفرديين والإحصاءات الخاصـــة بـــه حيث يوجد لكل فريق العديد من اللاعبين ولكل لاعب منهم الإحصاءات الخاصة به ويمكن تقسـيم عنصر TEAM إلى عناصر PLAYER ويوضع كل لاعب في مقطع مستوى الكتلة الخاصة بـــه

كما حدث مع العناصر السابقة ولكن هناك أسلوب أكثر فاعلية في تنظيم هـــذه العنـــاصر وهـــو استخدام الجداول وتبدو قواعد النمط التي تحقق هذا الغرض مثل ما يلي:

```
TEAM    {display: table}
TEAM_CITY {display: table-caption}
TEAM_NAME {display: table-caption}
PLAYER {display: table-row}
SURNAME {display: table-cell}
GIVEN_NAME {display: table-cell}
POSITION {display: table-cell}
GAMES {display: table-cell}
GAMES_STARTED {display: table-cell}
AT_BATS {display: table-cell}
RUNS {display: table-cell}
HITS {display: table-cell}
DOUBLES {display: table-cell}
TRIPLES {display: table-cell}
HOME_RUNS {display: table-cell}
RBI {display: table-cell}
STEALS {display: table-cell}
CAUGHT_STEALING {display: table-cell}
SACRIFICE_HITS {display: table-cell}
SACRIFICE_FLIES {display: table-cell}
ERRORS {display: table-cell}
WALKS {display: table-cell}
STRUCK_OUT {display: table-cell}
HIT_BY_PITCH {display: table-cell}
```

يتم تدعيم خصائص الجدول فقط في CSS Level 2 ولم يتـــم دعمـــها بواســـطة Internet Explorer 5.0 أو أي مستعرض آخر متاح في وقت كتابة هذا الكتاب وحالياً ســـنكتفي بجعـــل PLAYER وTEAM عناصر مستوى الكتلة لأن تنسيق الجدول لا يعمل حاليـــاً وســـنترك بقيـــة المستند بالتنسيق الافتراضي.

تلخيص

توضح تعليمات برمجة "٤-٢" ورقة النمط بعد الانتهاء منها ولا يوجد في أوراق نمط CSS بنيـــة إضافية أكثر من القواعد الفردية التي يضعها المستخدم. ما يلي هو كل القواعد التي تم تقديمـــــها بصورة منفصلة فيما سبق وإعادة ترتيب تلك القواعد لا يقدم أي إضافة جديدة طالما كل القواعـــد واضحة.

تعليمات برمجية ٤-٢ : ballstats.css

```
SEASON {font-size: 14pt; background-color: white;
       color: black; display: block}
YEAR {display: block; font-size: 32pt; font-weight: bold;
     text-align: center}
LEAGUE_NAME {display: block; text-align: center;
          font-size: 28pt; font-weight: bold}
DIVISION_NAME {display: block; text-align: center;
            font-size: 24pt; font-weight: bold}
TEAM_CITY {font-size: 20pt; font-weight: bold;
         font-style: italic}
TEAM_NAME {font-size: 20pt; font-weight: bold;
         font-style: italic}
TEAM {display: block}
PLAYER {display: block}
```

تكمل تعليمات البرمجة السابقة لتنسيق الأساسي لإحصاءات البيسبول ولكن مازال هناك بعض الأشياء التي يجب تنفيذها وتساعد المستعرضات التي تدعم تنسيق الجداول في تنفيذ تلك الأشـــياء ولكن يجب الاهتمام ببعض النقاط المذكورة فيما يلي:

- تقدم الأرقام بغير إشارة إلى ما تمثله ويتم التعرف على كل رقم بواسطة عنوان تســـمية مثل "RBI" و"At Bats ".
- لا يتم تضمن بيانات مثل متوسط ضرب الكرة التي يتم احتسابها من البيانات التـــي يتـــم عرضها.
- بعض العناوين قصيرة للغاية فمثلاً يكون من الأفضل إذا كان عنوان المســـتند "1998 Major League Baseball" بدلاً من "1998".

- تصعب قراءة المستند إذا تم تضمن كل لاعبي بطولة الدوري الرئيسي وفي هذه الحالـــة يكون من المناسب استخدام شيئاً مشابها لطريقـــة العـــرض المفصلـــة الممكـــن طيـــها المستخدمة في Internet Explorer للمستندات التي لا تملك ورقة نمط.
- من الأفضل فرز إحصاءات رامي الكرة بعيداً عن إحصاءات ضارب الكرة أثناء إعدادهم في roaster.

يمكن تنفيذ العديد من تلك النقاط بإضافة المزيد من المحتوى إلى المستند فمثلاً لتغيير العنـوان "1998" إلى "1998 Major League Baseball" كل ما يجب فعلة هو إعادة كتابـــة عنصـــر YEAR بالطريقة التالية:

1998 Major League Baseball.

يمكن إضافة عناوين التسمية لإحصاءات اللاعبين باستخدام لاعب وهمـــي فـــي بدايـــة كـــل roaster كما يلي:

```
<PLAYER>
 <SURNAME>Surname</SURNAME>
  <GIVEN_NAME>Given name</GIVEN_NAME>
  <POSITION>Postion</POSITION>
  <GAMES>Games</GAMES>
  <GAMES_STARTED>Games Started</GAMES_STARTED>
  <AT_BATS>At Bats</AT_BATS>
  <RUNS>Runs</RUNS>
  <HITS>Hits</HITS>
  <DOUBLES>Doubles</DOUBLES>
  <TRIPLES>Triples</TRIPLES>
  <HOME_RUNS>Home Runs</HOME_RUNS>
  <RBI>Runs Batted In</RBI>
  <STEALS>Steals</STEALS>
  <CAUGHT_STEALING>Caught Stealing</CAUGHT_STEALING>
  <SACRIFICE_HITS>Sacrifice Hits</SACRIFICE_HITS>
  <SACRIFICE_FLIES>Sacrifice Flies</SACRIFICE_FLIES>
  <ERRORS>Errors</ERRORS>
  <WALKS>Walks</WALKS>
  <STRUCK_OUT>Struck Out</STRUCK_OUT>
  <HIT_BY_PITCH>Hit By Pitch</HIT_BY_PITCH>
 </PLAYER>
```

مازال هناك بعض الأشياء مزعجة بشكل أساسي في هذا الأسلوب فالسنة هي ١٩٩٨ وليست 1998, Major League Baseball وعنوان التسمية At Bats ليس مثل عدد At Bats الفرق هنا يكون بين اسم الشيء والشيء نفسه ويمكن تشفير المزيد من الترميز كما يلي:

```
<TABLE_HEAD>
  <COLUMN_LABEL>Surname</COLUMN_LABEL>
  <COLUMN_LABEL>Given name</COLUMN_LABEL>
  <COLUMN_LABEL>Position</COLUMN_LABEL>
  <COLUMN_LABEL>Games</COLUMN_LABEL>
  <COLUMN_LABEL>Games Started</COLUMN_LABEL>
  <COLUMN_LABEL>At Bats</COLUMN_LABEL>
  <COLUMN_LABEL>Runs</COLUMN_LABEL>
  <COLUMN_LABEL>Hits</COLUMN_LABEL>
  <COLUMN_LABEL>Doubles</COLUMN_LABEL>
  <COLUMN_LABEL>Triples</COLUMN_LABEL>
  <COLUMN_LABEL>Home Runs</COLUMN_LABEL>
  <COLUMN_LABEL>Runs Batted In</COLUMN_LABEL>
  <COLUMN_LABEL>Steals</COLUMN_LABEL>
  <COLUMN_LABEL>Caught Stealing</COLUMN_LABEL>
  <COLUMN_LABEL>Sacrifice Hits</COLUMN_LABEL>
  <COLUMN_LABEL>Sacrifice Flies</COLUMN_LABEL>
  <COLUMN_LABEL>Errors</COLUMN_LABEL>
  <COLUMN_LABEL>Walks</COLUMN_LABEL>
  <COLUMN_LABEL>Struck Out</COLUMN_LABEL>
  <COLUMN_LABEL>Hit By Pitch</COLUMN_LABEL>
</TABLE_HEAD>
```

هذا يعيد ابتكار HTML ويعود بنا إلى نقطة استخدام الترميز للتنسيق أكثر من استخدامه كمعني وبالإضافة إلى ذلك فمازلنا نكرر المعلومات التي تحتويها أسماء العناصر وفي الواقع فالمستند كبير بما فيه الكفاية ولا توجد حاجة إلى جعله أكبر.

إضافة متوسط ضربات الكرة والمتوسطات الأخرى سهل. فقط قم بإضافة البيانات كعناصر إضافية فمثلا ما يلي هو اسم لاعب وقد تم حساب متوسط ضربات الكرة وon-base وslugging التي حققها ذلك اللاعب:

```
<PLAYER>
 <SURNAME>Malloy</SURNAME>
 <GIVEN_NAME>Marty</GIVEN_NAME>
 <POSITION>Second Base</POSITION>
 <GAMES>11</GAMES>
 <GAMES_STARTED>8</GAMES_STARTED>
 <ON_BASE_AVERAGE>.233</ON_BASE_AVERAGE>
 <SLUGGING_AVERAGE>.321</SLUGGING_AVERAGE>
 <BATTING_AVERAGE>.179</BATTING_AVERAGE>
 <AT_BATS>28</AT_BATS>
 <RUNS>3</RUNS>
 <HITS>5</HITS>
 <DOUBLES>1</DOUBLES>
 <TRIPLES>0</TRIPLES>
 <HOME_RUNS>1</HOME_RUNS>s
 <RBI>1</RBI>
 <STEALS>0</STEALS>
 <CAUGHT_STEALING>0</CAUGHT_STEALING>
 <SACRIFICE_HITS>0</SACRIFICE_HITS>
 <SACRIFICE_FLIES>0</SACRIFICE_FLIES>
 <ERRORS>0</ERRORS>
 <WALKS>2</WALKS>
 <STRUCK_OUT>2</STRUCK_OUT>
 <HIT_BY_PITCH>0</HIT_BY_PITCH>
</PLAYER>
```

تعتبر تلك المعلومات متوفرة لأنه يمكن حسابها من معلومات أخرى تم تضمينها بالفعل في تعليمات برمجة اللاعب فمثلاً متوسط ضرب الكرة يتم حسابه عن طريق قسمة رقم base hits على رقم at bats ويكون هذا HITS/AT_BATS. يجعل وفرة البيانات الحفاظ على وتحديث المستند صعب حيث يتطلب أي تغيير أو إضافة بسيطة في عنصر واحد إجراء تغييرات وإعادة حساب لمواقع متعددة.

المراد هنا هو لغة ورقة نمط مختلفة تتيح إضافة محتوى متداول محدد للعناصر وأداء تحويلات على محتوى العنصر الموجود وتلك اللغة هي Extensible Style Language (XSL).



تغطي الفصول ٥ و١٤ و١٥ "Extensible Style Language "XSL

لغة نمط CSS أسهل من لغة XSL وتعمل أيضاً بصورة جيدة في صفحات الويب الأساسـية والمستندات المباشرة وتعتبر لغة XSL معقدة بعض الشيء ولكنها أكثر قوة وتعتمد لغة XSL على تنسيق CSS وبالإضافة إلى ذلك فهي تقدم تحويلات المستند المصدر إلى نماذج متعددة يستطيع القارئ عرضها. من الأفضل البدء في علاج المشاكل التي قد تواجهها باستخدام CSS أثناء تصحيح مستند XSL للحصول على مرونة أكثر.

خلاصة

رأينا في هذا الفصل كيف يتم بناء مستند XML من البداية وتعلمنا ما يلي:

- كيفية اختبار البيانات المضمنة في مستند XML لتعريف العناصر.
- كيفية كتابة البيانات عن طريق الرموز باستخدام علامات XML التي تقوم بتعريفها.
- المزايا الإضافية التي تقدمها تنسيقات XML عن التنسيقات العادية.
- كيفية كتابة ورقة نمط توضح كيف يتم تنسيق وعرض المستند.

امتلأ هذا الفصل بتعليمات برمجية seat-of-the-pants/back-of-the-envelope ولقد تم كتابة المستند بدون أدنى اهتمام بالتفاصيل. نستعرض في الفصل القادم بعض الوسائل الإضافية لتضمين المعلومات في مستندات XML بما في ذلك السمات والتعليقات وتعليمات المعالجة ونستعرض أيضاً طريقة بديلة لتشفير إحصاءات البيسبول في XML.



الفصل

السمات والعلامات الفارغة وXSL

في هذا الفصل

- السمات في مقابل
- السمات
- العلامات الفارغة
- XSL

يمكن وضع تعليمات البرمجة لمجموعة من البيانات في مستند XML بعدد لا حصر له من الطرق، كما أنه لا يمكن الزعم بأن طريقة واحدة فقط هي الطريقة الصحيحة للقيام بمثل هذه العملية، بيد أن هناك بعض الطرق التي تتميز بكونها أكثر صحة من الطرق الأخرى بالإضافة إلى وجود بعض منها يتناسب مع مستخدمين معينين. ويتم في هذا الفصل اكتشاف حل مختلف لكتابة إحصائيات البيسبول بلغة XML مع نقل مثال البيسبول من الفصل السابق. وعلى وجه التحديد سوف يتم توجيه استخدام السمات لحفظ المعلومات والعلامات الفارغة لتحديد مواضع العناصر. ولأن CCS لا تعمل مع عناصر أقل من XML لهذه الصيغة فإنه سوف يتم في هذا الفصل عرض بعض المعلومات عن البديل الأقوى وهو لغة ورقة النمط XSL.

السمات

تم في الفصل السابق تصنيف كل البيانات على أساس اسم أحد العلامات أو مكونات أحد العناصر. وتعد هذه الطريقة من الطرق البسيطة التي لا تحتوي على الكثير من التعقيدات، غير أنها ليست الطريقة الوحيدة. وكما أن عناصر HTML لها سماتها المميزة، فإن عناصر XML لها أيضاً مميزاتها. والسمة ما هي إلا زوج يتكون من الاسم والقيمة المرتبطة بأحد العناصر.

والصيغ الخاصة بسمات HTML غير جديدة على القارئ، فعلى سبيل المثال هذه العلامة:

```
<IMG SRC=cup.gif WIDTH=89 HEIGHT=67 ALT"="Cup of coffee"">
```

لها أربعة سمات: سمة SRC والتي قيمتها هي cup.gif، وسمة WIDTH والتي قيمتها ٨٩ وسمة HIEGHT والتي قيمتها ٦٧، وسمة ALT والتي قيمتها Cup of coffee. غير أن قيم سمات XML، على العكس من قيم سمات HTML، يجب دائماً أن يتم وضعها بين علامتي اقتباس، كما أن علامات البداية يجب أن يكون لها علامات نهاية متناسبة معها. ومن ثم فإن صيغة XML المساوية لها تكون:

```
<IMG SRC"="cup.gif" WIDTH"="89"" HIEGHT""="67"" ALT"="Cup of
coffee"">
</IMG>
```



هناك فرق آخر بين HTML وXML وهو أن الأخير لا يحدد معنى معيناً لعلامة IMG وسماتها. وليس هناك أي ضمان على أن مستعرض XML سوف يقوم بترجمة هذه العلامة كإرشاد لتحميل وعرض الصور من ملف cup.gif.

يمكن تطبيق صيغة السمة على مثال البيسبول بسهولة تامة مما يجعل الترمز أكثر اختصاراً. فعلى سبيل المثال بدلاً من وضع عنصر تابع YEAR، يحتاج عنصر SEASON إلى سمة YEAR.

```
<SAESON YEAR"="1998"">
</SEASON>
```

ومن ناحية أخرى، فإن LEAGE يجب أن يكون تابعاً لعنصر SEASON بدلاً من أن تكــون سمة. والسبب الأول هو أنه هناك بطولتان في الموسم الواحد. وفي أي وقت، فإن هناك احتمـــال أن يكون هناك أكثر من عنصر تابع يتم استدعاؤه كما يجب أن تكون أسماء السمات متميزة داخل العنصر، ومن ثم فإنه يجب على سبيل المثال ألا يتم كتابة عنصر SEASON بهذه الطريقة:

```
<SEASON YEAR=""1998" League="National"League="American">
</SEASON>
```

والسبب الثاني في كون LEAGUE عنصر تابع وليس سمة هو أن له بنية فرعية، فهو مقســم تقسيم فرعي إلى عناصر DIVISION. وقيم السمات عبارة عن نصوص غير مفصلة. تســتطيع عناصر XML وضع التعليمات الخاصة بالبنية بشكل ملائم، بينما لا تستطيع قيــم الســمة القيــام بذلك.

غير أن اسم البطولة دائماً ما يكون نصاً غير مفصل وليس له بنية، وهناك اسم واحد فقط لكل بطولة لذلك فإن عناصر البطولة يمكن أن يكون لها سمة NAME بــدلاً مــن العنصــر التــابع LEAGUE_ELEMENT:

```
<LEAGUE NAME="NATIONAL LEAGUE">
</LEAGUE>
```

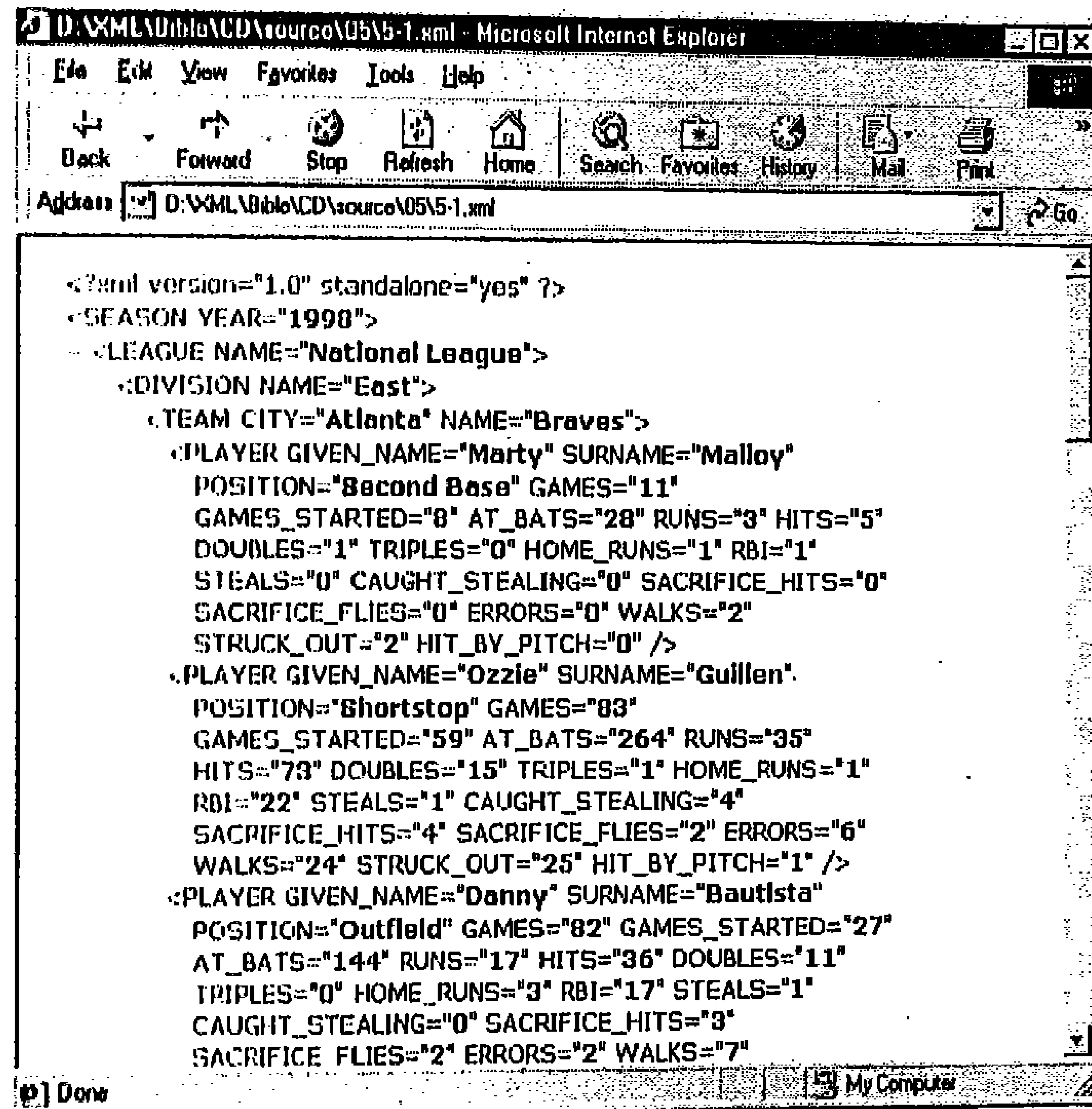
ولأن السمة تكون مرتبطة بالعنصر الخاص بها أكثر من ارتباط العنصر التــابع بــه، فــإن استخدام NAME بدلاً من LEAGUE_NAME كاسم للسمة لن يتسبب في حدوث أية مشاكل. يمكن أن يكون للفرق والأقسام سمات NAME بدون أدنى احتمال من تداخلها مع اسم البطولــــة. ولأن العلامة يمكن أن تحتوي على أكثر من سمة بما أن السمات لها أسماء مختلفة فإنه يمكــن جعــل البلدة الخاصة بالفريق سمة أيضاً كما هو موضح في المثال التالي:

```
<LEAGUE NAME="American League">
        <DIVISION NAME="East">
                <TEAM NAME="Orioles" CITY="Baltimore"></TEAM>
                <TEAM NAME="Red Sox" CITY="Boston"></TEAM>
                <TEAM NAME="Yankees" CITY="New York"></TEAM>
                <TEAM NAME="Devil Rays" CITY="Tampa
        Bay"></TEAM>
                <TEAM NAME="Blue Jays" CITY="Toronto"></TEAM>
        </DIVISION>
</LEAGUE>
```

وسوف يكون للاعبين عدة سمات إذا كانت هناك رغبة في جعل كل إحصاء يعبر عن سـمة. فعلى سبيل المثال هاهي إحصاءات جو جيراردي لعام ١٩٩٨ على شكل سمات:

```
<PLAYER GIVEN_NAME="Joe" SURNAME="Girardi"
        GAMES="78" AT_BATS="254" RUNS="31" HITS="70"
        DOUBLES="11" TRIPLES="4" HOME_RUNS="3"
        RUNS_BATTED_IN="31" WALKS="14" STRUCK_OUT="38"
        STOLEN_BASES="2" CAUGHT_STEALING="4"
        SACRIFICE_FLY="1" SACRIFICE_HIT="8"
        HIT_BY_PITCH="2">
  </PLAYER>
```

وتستخدم تعليمات البرمجة ٥-١ هذا الشكل الجديد من السمات لعمل مستندXML كامل يحتوي على إحصائيات موسم ١٩٩٨ لبطولة البيسبول. وهي تعرض نفس المعلومات "مثل أنها تتكـــون من بطولتين وست مجموعات وثلاثين فريقاً وكل فريق يحتوي على تسعة لاعبيـــن" كمـــا هـــو موضح في تعليمات البرمجة في الفصل السابق. غير أن الفرق الوحيد يظـــهر فـــي الأســـلوب. ويوضح الشكل ٥-١ هذا المستند بعد تحميله في Internet Explorer 5.0 بدون ورقة النمط.



الشكل ٥-١ إحصاءات بطولة البيسبول لعام ١٩٩٨ باستخدام السمات لأغلب المعلومات.

تعليمات برمجة ٥-١ : مستند XML كامل والذي يستخدم لحفظ إحصائيات البيسبول.

```
<?xml version="1.0" standalone="yes"?>
<SEASON YEAR=1998>
 <LEAGUE NAME="National League">
  <DIVISION NAME="East">
   <TEAM CITY="Atlanta" NAME="Braves">
    <PLAYER GIVEN_NAME="Marty" SURNAME="Malloy"
     POSITION="Second Base" GAMES="11" GAMES_STARTED="8"
     AT_BATS="28" RUNS="3" HITS="5" DOUBLES="1"
     TRIPLES="0' HOME_RUNS="1" RBI="1" STEALS="0"
     CAUGHT_STEALING="0" SACRIFICE_HITS="0"
     SACRIFICE_FLIES="0" ERRORS="0" WALKS="2"
     STRUCK_OUT="2" HIT_BY_PITCH="0">
    </PLAYER>
    <PLAYER GIVEN_NAME="Ozzie" SURNAME="Guillen"
     POSITION="Shortstop" GAMES="83" GAMES_STARTED="59"
     AT_BATS="264" RUNS="35" HITS="73" DOUBLES="15"
     TRIPLES="1" HOME_RUNS="1" RBI="22" STEALS="1"
     CAUGHT_STEALING="4" SACRIFICE_HITS="4"
     SACRIFICE_FLIES="2" ERRORS="6" WALKS="24"
     STRUCK_OUT="25" HIT_BY_PITCH="1">
    </PLAYER>
    <PLAYER GIVEN_NAME="Danny" SURNAME="Bautista"
      POSITION="Outfield" GAMES="82" GAMES_STARTED="27"
      AT_BATS="144" RUNS="17" HITS="36" DOUBLES="11"
      TRIPLES="0" HOME_RUNS="3" RBI="17" STEALS="1"
      CAUGHT_STEALING="0" SACRIFICE_HITS="3"
      SACRIFICE_FLIES="2" ERRORS="2" WALKS="7"
      STRUCK_OUT="21" HIT_BY_PITCH="0">
    </PLAYER>
    <PLAYER GIVEN_NAME="Gerald" SURNAME="Williams"
```

```
POSITION="Outfield" GAMES="129" GAMES_STARTED="51"
AT_BATS="266" RUNS="46" HITS="81" DOUBLES="18"
TRIPLES="3" HOME_RUNS="10" RBI="44" STEALS="11"
CAUGHT_STEALING="5" SACRIFICE_HITS="2"
SACRIFICE_FLIES="1" ERRORS="5" WALKS="17"
STRUCK_OUT="48" HIT_BY_PITCH="3">
</PLAYER>
<PLAYER GIVEN_NAME="Tom" SURNAME="Glavine"
POSITION="Starting Pitcher" GAMES="33"
GAMES_STARTED="33" WINS="20" LOSSES="6" SAVES="0"
COMPLETE_GAMES="4" SHUT_OUTS="3" ERA="2.47"
INNINGS="229.1" HOME_RUNS_AGAINST="13"
RUNS_AGAINST="67" EARNED_RUNS="63" HIT_BATTER="2"
WILD_PITCHES="3" BALK="0" WALKED_BATTER="74"
STRUCK_OUT_BATTER="157">
</PLAYER>
<PLAYER GIVEN_NAME="Javier" SURNAME="Lopez"
POSITION="Catcher" GAMES="133" GAMES_STARTED="124"
AT_BATS="489" RUNS="73" HITS="139" DOUBLES="21"
TRIPLES="1" HOME_RUNS="34" RBI="106" STEALS="5"
CAUGHT_STEALING="3" SACRIFICE_HITS="1"
SACRIFICE_FLIES="8" ERRORS="5" WALKS="30"
STRUCK_OUT="85" HIT_BY_PITCH="6">
</PLAYER>
<PLAYER GIVEN_NAME="Ryan" SURNAME="Klesko"
POSITION="Outfield" GAMES="129" GAMES_STARTED="124"
AT_BATS="427" RUNS="69" HITS="117" DOUBLES="29"
TRIPLES="1" HOME_RUNS="18" RBI="70" STEALS="5"
CAUGHT_STEALING="3" SACRIFICE_HITS="0"
SACRIFICE_FLIES="4" ERRORS="2" WALKS="56"
STRUCK_OUT="٦٦" HIT_BY_PITCH="٣">
</PLAYER>
<PLAYER GIVEN_NAME="Andres" SURNAME="Galarraga"
```

```
    POSITION="First Base" GAMES="135" GAMES_STARTED="151"
    AT_BATS="555" RUNS="103" HITS="169" DOUBLES="27"
    TRIPLES="1" HOME_RUNS="44" RBI="121" STEALS="7"
    CAUGHT_STEALING="6" SACRIFICE_HITS="0"
    SACRIFICE_FLIES="5" ERRORS="11" WALKS="63"
    STRUCK_OUT="146" HIT_BY_PITCH="25">
   </PLAYER>
   <PLAYER GIVEN_NAME="Wes" SURNAME="Helms"
    POSITION="Third Base" GAMES="7" GAMES_STARTED="2"
    AT_BATS="13" RUNS="2"0 HITS="4" DOUBLES="1"
    TRIPLES="0" HOME_RUNS="1" RBI="2" STEALS="0"
    CAUGHT_STEALING="0" SACRIFICE_HITS="0"
    SACRIFICE_FLIES="0" ERRORS="1" WALKS="0"
    STRUCK_OUT="4" HIT_BY_PITCH="0">
   </PLAYER>
  </TEAM>
  <TEAM CITY="Florida" NAME="Marlins">
  </TEAM>
  <TEAM CITY="Montreal" NAME="Expos">
  </TEAM>
  <TEAM CITY="New York" NAME="Mets">
  </TEAM>
  <TEAM CITY="Philadelphia" NAME="Phillies">
  </TEAM>
</DIVISION>
<DIVISION NAME="Central">
  <TEAM CITY="Chicago" NAME="cubs">
  </TEAM>
  <TEAM CITY="Cincinnati" NAME="Reds">
  </TEAM>
  <TEAM CITY="Houston" NAME="Astros">
  </TEAM>
  <TEAM CITY="Milwaukee" NAME="Brewers">
  </TEAM>
```

```
<TEAM CITY="Pittsburgh" NAME="Pirates">
</TEAM>
<TEAM CITY="St. Louis" NAME="Cardinals">
</TEAM>
</DIVISION>
<DIVISION NAME="west">
<TEAM CITY="Arizona" NAME="Diamondbacks">
</TEAM>
<TEAM CITY="colorado" NAME="rockies">
</TEAM>
<TEAM CITY="Los Angeles" NAME="Dodgers">
</TEAM>
<TEAM CITY="San Diego" NAME="Padres">
</TEAM>
<TEAM CITY="an Francisco" NAME="giants">
</TEAM>
</DIVISION>
</LEAGUE>
<LEAGUE NAME="American League">
<DIVISION NAME="East">
<TEAM CITY="Baltimore" NAME="Orioles">
</TEAM>
<TEAM CITY="Boston" NAME="Red Sox">
</TEAM>
<TEAM CITY="New York" NAME="Yankees">
</TEAM>
<TEAM CITY="Tampa Bay" NAME="Devil Rays">
</TEAM>
<TEAM CITY="Toronto" NAME="Blue Jays">
</TEAM>
</DIVISION>
<DIVISION NAME="Central">
<TEAM CITY="Chicago" NAME="White Sox">
</TEAM>
```

```
    <TEAM CITY="Kansas City" NAME="Royals">
    </TEAM>
    <TEAM CITY="Detroit" NAME="Tigers">
    </TEAM>
    <TEAM CITY="Cleveland" NAME="Indians">
    </TEAM>
    <TEAM CITY="Minnesota" NAME="Twins">
    </TEAM>
  </DIVISION>
  <DIVISION NAME="West">
    <TEAM CITY="Anaheim" NAME="Angels">
    </TEAM>
    <TEAM CITY="Oakland" NAME="Athletics">
    </TEAM>
    <TEAM CITY="Seattle" NAME="Mariners">
    </TEAM>
    <TEAM CITY="Texas" NAME="Rangers">
    </TEAM>
  </DIVISION>
 </LEAGUE>
</SEASON>
```

وتستخدم تعليمات برمجة ٥-١ السمات للمعلومات عن اللاعب فقط. أما تعليمات برمجة ٤-١ فإنها تستخدم محتوى العنصر. بالإضافة إلى وجود بعض المناهج الوسيطة بين هذين المنهجين، فيمكن على سبيل المثال جعل اسم اللاعب جزءاً من محتوى العنصر، مع ترك باقي الإحصائيات كسمات كما يلي:

```
 <P>
On Tuesday <PLAYER GAMES="78" AT_BATS="254" RUNS="31"
HITS="٧٠" DOUBLES="11" TRIPLES="4" HOME_RUNS="3"
RUNS_BATTED_IN="31" WALKS="14" STRIKE_OUTS="38"
STOLEN_BASES="2" CAUGHT_STEALING="4"
SACRIFICE_FLY="1" SACRIFICE_HIT="8"
```

```
HIT_BY_PITCH="2">Joe Girardi</PLAYER> struck out twice
 and...
</P>
```

وسوف يؤدي ذلك إلى تضمين اسم جو جيراردي في النص في أحد الصفحات بينما تكون الإحصائيات الخاصة به متاحة للقراء الراغبين في مزيد من المعلومات، كحواشي سفلية متشعبة أو كتعريف للأدوات. دائماً ما يكون هناك طريقة واحدة لوضع تعليمات البرمجة لنفس البيانات. وتعتمد الطريقة التي يتم اختيارها على احتياجات التطبيق المستخدم.

السمات في مقابل العناصر

ليس هناك قواعد محددة فيما يخص استخدام العناصر التابعة. وبوجه عام يتم اختيار الطريقة التي تتناسب مع التطبيق. ومع الخبرة سوف تكتسب مهارة الإحساس بالأوقات التي تكون فيها السمات أسهل من العناصر التابعة، والعكس صحيح، وحتى هذه المرحلة فإن هناك قاعدة مؤكدة وهي أن البيانات نفسها يجب أن يتم تخزينها في عناصر، كما يجب تخزين معلومات البيانات في سمات. غير أنه يتم وضع المعلومات في عناصر عند الشك.

وللتميز بين البيانات والبيانات المفصلة يجب على المستخدم أن يقوم بسؤال نفسه ما إذا كان أحد الأشخاص الذين يقرؤون هذا المستند سيحتاج إلى أن يرى معلومة معينة. وإذا كانت الإجابة بنعم فإن ذلك قد يعني أن المعلومات تنتمي إلى عنصر تابع، أما إذا كانت الإجابة بلا فإن ذلك قد يعني أن المعلومات قد تنتمي إلى السمات. وإذا كانت كل العلامات قد تم اقتباسها من المستند بالإضافة إلى كل السمات، فإنه يجب الحفاظ على كل المعلومات الأساسية. وتعد السمات أماكن جيدة لوضع أرقام تعريف وURL والمراجع والبيانات الأخرى التي تكون غير متاحة للقارئ في الوقت الحالي، غير أنه هناك بعض الاستثناءات للقاعدة الأساسية التي تحول دون تخزين البيانات المفصلة على هيئة سمات.وهي:

- السمات لا تستطيع التحكم في البنية بشكل جيد.
- العناصر التي تسمح بتضمين المعلومات الأكثر تفصيلاً "معلومات عن معلومات عن المعلومات".
- دائماً ما يحدث خلاف في وجهات النظر على ما هي المعلومات المفصلة.
- دائماً ما تكون المعلومات قابلة للزيادة مع الزمن.

بنية البيانات التفصيلية

من أحد المبادئ الهامة التي يجب وضعها في الاعتبار هي أن العناصر يمكن أن يكون لها بنية، بينما السمات لا يكون لها بنية. وهذا من شأنه أن يجعل العناصر أكثر مرونة إلى حد بعيد، كما أنه من شأنه أن يجعلك تقتنع بوضع تعليمات برمجة للبيانات المفصلة على أنها عناصر تابعة.فعلى سبيل المثال افترض أنك تقوم بكتابة أحد الأبحاث وتريد تضمين مصدراً لأحد الحقائق. فإن ذلك سيبدو هكذا:

```
<FACT SOURCE="The Biographical History of Baseball,
Donald Dewey and Nicholas Acocella (New York: Carroll &amp;
Graf Publishers, Inc. 1995) p. 169">
 Josh Gibson is the only person in the history of baseball to
 hit a pitch out of Yankee Stadium.
</FACT>
```

ومن الواضح أن المعلومات التي تتناول كتاب تاريخ البيسبول والمؤلفين ومكان طباعة هذا الكتاب والمطبعة وسنة الطباعة، وصفحة ١٦٩، كل هذه عبارة عن بيانات مفصلة، حيث أنها ليست الحقيقة نفسها، ولكنها عبارة عن بعض المعلومات الأساسية عن الحقيقة غير أن سمة SOURCE تحتوي على العديد من البنيات الداخلية. ومن المفيد تنظيم المعلومات على هذا النحو:

```
<SOURCE>
 <AUTHOR>Donald Dewey</AUTHOR>
 <AUTHOR>Nicholas Acocella</AUTHOR>
 <BOOK>
  <TITLE>The Biographical History of Baseball</TITLE>
  <PAGES>169</PAGES>
  <YEAR>1995</YEAR>
 </BOOK>
</SOURCE>
```

وبالإضافة إلى ذلك فإن استخدام العناصر بدلاً من السمات يجعل عملية إضافة المعلومات الجديدة مثل عناوين البريد الإلكتروني للكاتب وURL عند إمكانية إيجاد نسخة إلكترونية من المستند أو عنوان أو أي عنصر من عناصر الصحيفة أو أي شئ قد يبدو مهماً يتم بطريقة مباشرة.

وتعد التواريخ أيضا من الأمثلة الشائعة فيما يخص هذه الحالة. حيث أن أحد البيانات المفصلة عن المقالات المدرسية هي التاريخ الذي تم فيه استلام المقال لأول مرة. ويعد هذا في غاية

الأهمية عند الرغبة في معرفة الأسبقية. ومن السهل تضمين سمة DATE في علامة ARTICLE كما يلي:

```
<ARTICLE DATE="06/28/1969">
 Polymerase Reactions in Organic Compounds
</ARTICLE>
```

غير أن سمة DATE لها بنية فرعية يتم تحديدها بالعلامة/. وإخراج هذه البنية من قيمة السمة يتسم بالصعوبة عن قراءة العناصر التابعة لعنصر DATE، كما هو موضح فيما يلي:

```
<DATE>
  <YEAR>1969</YEAR>
  <MONTH>06</MONTH>
  <DAY>28</DAY>
</DATE>
```

فعلى سبيل المثال، من السهل تنسيق اليوم والتاريخ بشكل غير مرئي في CSS أو XSL مما يجعل السنة فقط هي التي تظهر. فمثلاً باستخدام CSS:

```
YEAR {display: inline}
MONTH {display: none}
DAY {display: none}
```

غير أنه إذا تم تخزين DATE على هيئة سمة، فإن ذلك سوف يعوق عملية الوصول إلى أحد الأجزاء منها.مما سوف يلزمك بكتابة برنامج منفصل في إحدى لغات البرمجة مثل JAVA أو ECMAScript التي تستطيع تحليل التنسيق الخاص بالتاريخ.ولذلك فإنه من الأسهل استخدام العناصر التابعة وأدوات XML القياسية.

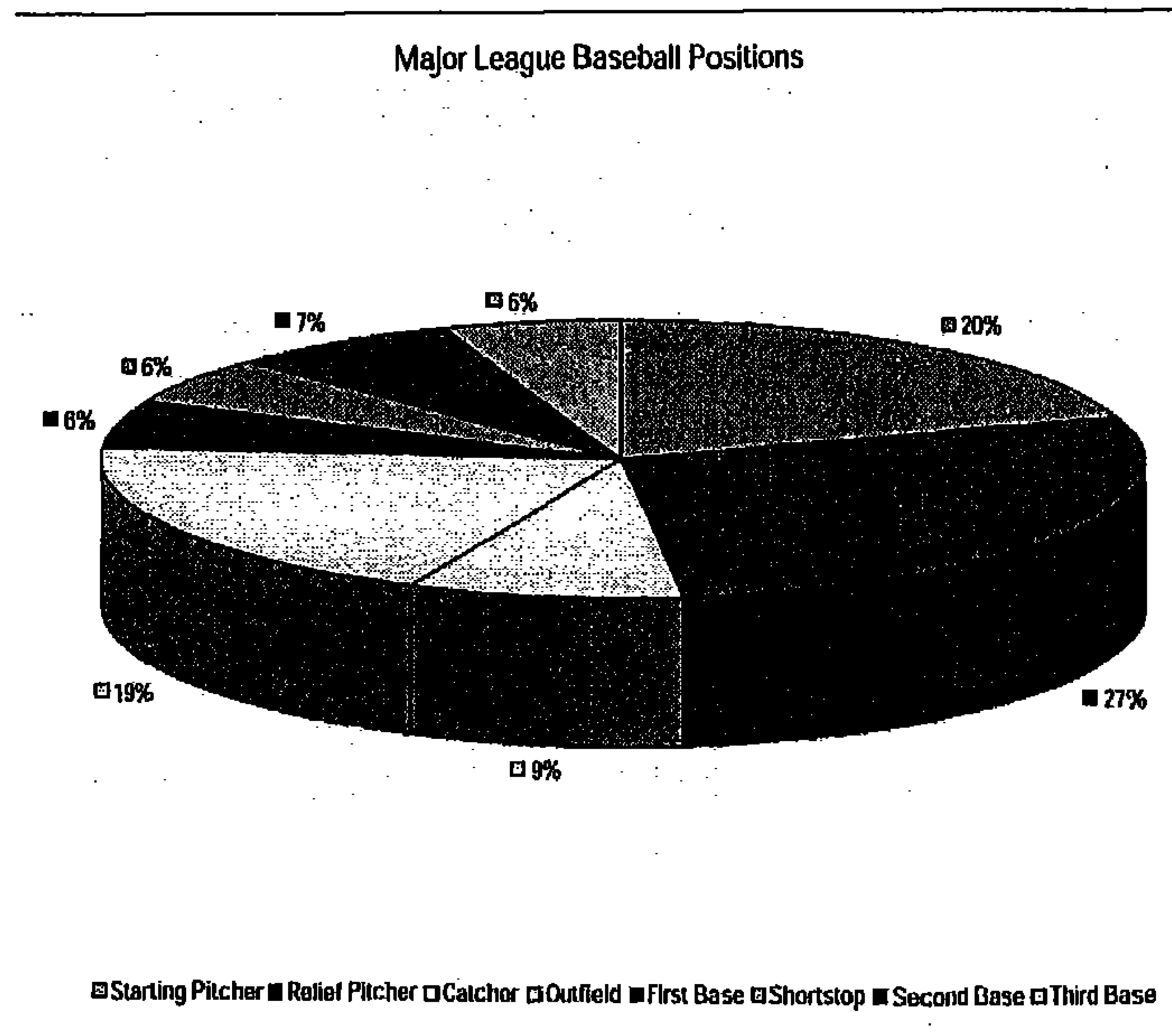
بالإضافة إلى ذلك فإن صيغة السمة تكون غير واضحة. حيث لا تدل الصيغة على التاريخ بدقة حيث سوف يقوم القراء بفهم هذه التواريخ كل على طريقته حتى إذا استطاع المحلل فهم أحد التنسيقات، فإنه ليس هناك ما يضمن أن مدخلي البيانات سوف يدخلونها على النحو الصحيح. وعلى النقيض من ذلك فإن XML يتسم بالوضوح.

وفي النهاية، فإن استخدام توابع DATE بدلاً من السمات يتيح لك ربط أكثر من تاريخ واحد بالعنصر، فعلى سبيل المثال فإنه دائماً ما يتم إرجاع المقالات المدرسية إلى الكاتب للمراجعة، ومن المهم في مثل هذه الحالات أن تلاحظ تاريخ استلام هذه المقالات على النحو التالي:

```
<ARTICLE>
        <TITLE>
        Maximum Projectile Velocity in an Augmented Railgun
        </TITLE>
        <AUTHOR>Elliotte Harold</AUTHOR>
```

```
        <AUTHOR>Bruce Bukiet</AUTHOR>
        <AUTHOR>William Peter</AUTHOR>
        <DATE>
        <YEAR>1992</YEAR>
        <MONTH>10</MONTH>
        <DAY>29</DAY>
        </DATE>
        <DATE>
        <YEAR>1993</YEAR>
        <MONTH>10</MONTH>
        <DAY>26</DAY>
        </DATE>
</ARTICLE>
```

مثال آخر، وهو سمة ALT الخاصة بعلامة IMG في HTML، وهي محدودة سلسلة واحدة لنص. ورغم أن الصورة في مثل هذه الحالة تغني عن نص يحتوي على آلاف الكلمات، فإن لمستخدم قد يحتاج إلى إدخال نص مميز بدلاً من IMG. فعلى سبيل المثال عند النظر إلى هذا لرسم البياني الموجود في الشكل ٥-٢:



الشكل ٥-٢ توزيع المراكز في بطولة البيسبول الكبرى.

فإنه باستخدام سمة ALT يكون أفضل وصف لهذه الصورة هو:

```
<IMG SRC="05021.gif"
```

```
   ALT="Pie Chart of Positions in Major League Baseball"
   WIDTH="819" HEIGHT="623">
</IMG>
```

غير أنه مع استخدام العنصر التابع ALT فإن هذه العملية سوف تتميز بالمرونة لأنـــه يمكــن تضمين نص مميز، فيمكن مثلاً وضع أحد الجداول التي تشتمل على الأرقام الهامة بدلاً من هــذا الرسم البياني.

```
<IMG SRC="05012gif" WIDTH="819" HEIGHT="623">
 <ALT>
  <TABLE>
   <TR>
    <TD>Starting Pitcher</TD> <TD>242</TD> <TD>20%</TD>
   </TR>
   <TR>
    <TD>Relief Pitcher</TD> <TD>336</TD> <TD>27%</TD>
   </TR>
   <TR>
    <TD>Catcher</TD> <TD>104</TD> <TD>9%</TD>
   </TR>
   <TR>
    <TD>Outfield</TD> <TD>235</TD> <TD>19%</TD>
   </TR>
   <TR>
    <TD>First Base</TD> <TD>67</TD> <TD>6%</TD>
   </TR>
   <TR>
    <TD>Shortstop</TD> <TD>67</TD> <TD>6%</TD>
   </TR>
   <TR>
    <TD>Second Base</TD> <TD>88</TD> <TD>7%</TD>
   </TR>
   <TR>
    <TD>Third Base</TD> <TD>67</TD> <TD>6%</TD>
   </TR>
```

```
   </TABLE>
  </ALT>
</IMG>
```

حتى أنه من الممكن أن تقوم بتوفير تعليمات البرمجة الفعلية لكل مـــن Postscripth أوSVG أو VML لاستعادة الصورة عندما تكون الصورة النقطية غير متاحة.

البيانات الأكثر تفصيلاً

يتيح استخدام العناصر للبيانات المفصلة استخدام البيانات الأكثر تفصيلاً، وهو مـــا يقصـــد بـــه معلومات عن معلومات عن معلومات. فعلى سبيل المثال فإن مؤلف أحد القصائد يمكن أن يعتـــبر من البيانات المفصلة عن القصيدة. إلا أن اللغة التي تمت بها كتابة مثل هذا الاسم تعتبر بيانـــات عن البيانات المفصلة لهذه القصيدة. وهذا لا يعد شيئاً بسيطاً خاصة في اللغات التي لا تنتمي إلـــى اللغات الأوروبية.فمثلاً الشاعر الذي كتب الإلياذة والأوديسة هل هو HOMER أم Vmho فـــإذا كنت تستخدم العناصر، يصبح من السهل كتابة:

```
<POET LANGUAGE="English">Homer</POET>
<POET LANGUAGE="Greek">Vmho§</POET>
```

ولكن إذا كان POET هو سمة وليس عنصراً تابعاً فإن ذلك سوف يؤدي إلى ما يلي:

```
<POEM POET="Homer" POET_LANGUAGE="English"
 POEM_LANGUAGE="English">
  Tell me, O Muse, of the cunning man.
</POEM>
```

وسوف تكون أكثر حجماً إذا كانت هناك رغبة في تضمين كل من الاسم باللغـــة الإنجليزيـــة واليونانية.

```
<POEM POET_NAME_1="Homer" POET_LANGUAGE_1="English"
 POET_NAME_2="Vmho" POET_LANGUAGE_2="greek"
 POEM_LANGUAGE="English">
  Tell me, O Muse, of the cunning man...
</POEM>
```

بياناتك المفصلة هي بيانات أحد الأشخاص الآخرين

يعتمد كون البيانات مفصلة أو غير مفصلة على القارئ نفسه وعلى السبب الذي جعله يقرأ هـــذه البيانات. فمثلاً إذا كنت تقرأ أحد المقالات في أحد الصحف المدرسية فإن الكاتب سوف يتنـــاول

الموضوع تناول سطحي.غير أنك إذا كنت ضمن لجنة لتفحص الأمر لمعرفة من قد قام بالنشر ومن لم يقم بالنشر فإن أسماء الكتاب وعدد المقالات سوف تكون أهم من محتويات المقالات نفسها.

في الواقع قد تكون أحد البيانات المهمة الآن غير مهمة بعد فترة، والعكس صحيح. ويمكن استخدام ورقة النمط لإخفاء البيانات غير الهامة اليوم، ثم تقوم بتغييرها لإظهار هذه البيانات فيما بعد. غير أنه من الصعب إظهار البيانات التي تم تخزينها على هيئة سمة في وقت تالي، ويتطلب هذا عادة إعادة كتابة النص نفسه بدلاً من تغيير ورقة النمط.

العناصر أكثر امتداداً

تكون السمات مناسبة عند الحاجة إلى نقل كلمة أو كلمتين إلى المعلومات التي لا تخضع للبنية.ولن تكون هناك حاجة في مثل هذه الحالة إلى العنصر التابع.غير أنه قد تكون هناك حاجة إليه في المستقبل.

وقد تحتاج إلى تخزين اسم كاتب المقال كما أنك قد لا تحتاج إلى التمييز بين الاسم الأول والاسم الأخير.غير أنك قد لا تحتاج إلى تخزين الأسماء الأولى والأخيرة وعناوين البريد الإلكتروني والهيئات وURL وما إلى ذلك.وإذا تم تخزين اسم كاتب المقال كعنصر، فإن عملية إضافة عناصر تابعة لتضمين هذه المعلومات الإضافية.

وعلى الرغم من أن أي تغيير سوف يحتاج بعض المراجعة على النصوص وأوراق النمط البرامج المتصلة به، فإن تغيير أحد العناصر البسيطة إلى شجرة العناصر يعد أمراً سهلاً عند مقارنته بعملية جعل إحدى السمات شجرة للعناصر. ولكن إذا استخدمت إحدى السمات فأن هذا سوف يؤدي إلى بعض المشاكل، حيث أنه من غير السهل مد بناء جملة السمة أكثر من المنطقة التي قد تم تحديدها لها من البداية.

الأوقات الملائمة لاستخدام السمات

بعدما تم إيضاح ،بكل الطرق الممكنة، كيفية أن استخدام العناصر أفضل من استخدام السمات، لا يمكن إغفال أن السمات يتم استخدامها في بعض الأوقات. فالسمات مناسبة لأية بيانات لا تخضع لبنية فرعية من غير المحتمل أن يراها القارئ، وأحد الأمثلة على ذلك هو السمات HEIGHT وWIDTH التي تخص IMG. ورغم أن قيم هذه السمات قد تتغير إذا حدث تغيير في الصورة، إلا أن السمة سوف تكون سلسلة قصيرة جداً من النص. وتتميز كل من HEIGHT و WIDTH بأنها كميات ذات بعد واحد، لذلك فإن كل منهما يصلح لكي يكون سمة.

وبالإضافة إلى ذلك فإن السمات تلائم المعلومات البسيطة الخاصة بالمستند والتي ليس لها أي علاقة بمحتواه. فمثلاً من المفيد تعيين سمة ID لكل عنصر.وتعد هذه سلسلة فريدة تخص عنصراً واحداً فقط من المستند. ويمكن استخدام هذه السلسلة لعدة مهام بما في ذلك الربط بين أحد العناصر حتى إذا تغيرت العناصر مع تغير المستند عبر الوقت مثل:

```
<SOURCE ID="S1">
  <AUTHOR ID="A1">Donald Dewey</AUTHOR>
  <AUTHOR ID="A2">Nicholas Acocella</AUTHOR>
  <BOOK ID="B1">
    <TITLE ID="B2">
      The Biographical History of Baseball
    </TITLE>
    <PAGES ID="B3">169</PAGES>
    <YEAR ID="B4">1995</YEAR>
  </BOOK>
</SOURCE>
```

فإن سمات ID تقوم بعمل روابط بعناصر معينة في المستند المتاح. وبهذه الطريقة يمكن أن تقوم بنفس الوظيفة التي تقوم بها سمةNAME الخاصة بعنصر HTML وهوA. كما أنه هناك بيانات أخرى تقوم بالربط مثل HREF وSRC لسحب الصور والبيانات الثنائية وما إلى ذلك وتعمل بشكل جيد كسمات.

المرجع يتم عرض المزيد من الأمثلة في هذا الشأن في الفصل ١٦ في الفصل ١٧.

كما تستخدم السمات دائماً لتخزين المعلومات المحددة عن شكل البيانات. فمثلاً إذا كانت هناك رغبة في جعل أحد عناصر TITLE سميكة ومائلة، فإنه يتم كتابة:

```
<TITLE style="font-style: italic">Significant Others</TITLE>
```

ويمكن ذلك المعلومات الخاصة بالشكل من تضمينها من غير تغيير شجرة البنية الخاصة بالمستند. وتقوم هذه العملية بإعطاء من يكتب المستند تحكماً أكثر عند عدم القدرة على إضافة المزيد من العناصر إلى مجموعة العلامة. فعلى سبيل المثال فإن الويب الأساسية في أحد المواقع قد تحتاج إلى استخدام DTD معين مع عدم السماح لأحد بتغييره. غير أنه قد يحتاج إلى عمل بعض التعديلات البسيطة. ويمكن استخدام هذه الطريقة مع بعض القيود، وإلا فإنك سوف تواجه نفس المشاكل الموجودة في HTML والتي نحاول تجنبها في XML حيث يتداخل التنسيق مع المعنى ولا يمكن الحفاظ على النص.

والسبب الأخير في استخدام السمات هو الحفاظ على التوافق مع HTML إلـــى الحـــد الـــذي تستخدم معه العلامات التـــي تبـــدو متشـــابهة مـــع علامـــات HTMLمثـــل <IMG> و<P> و<TD>.كما يمكن توظيف سمات HTML القياسية لهذه العلامات. ولهذه العملية ميزتان حيـــث تمكن المستعرضات من تحليل وعرض النص، بالإضافة إلى كونها معروفة أكثر إلى من يقـــوم بكتابة المستند.

العلامات الفارغة

تم عرض أحد المناهج التي تعارض استخدام السمات في الفصل السابق، غير أنه من الممكـــن أن يتم تخزين كل المعلومات فيها. ولكن بوجه عام هذه العملية غير محبذة حيـــث أن تخزيـــن كـــل المعلومات في محتوى العنصر أسهل عند التطبيق. ويقوم هذا الجزء بعرض إمكانيـــة اســـتخدام السمات للتوضيح فقط.

وبما أنك تعلم أن العنصر ليس له محتوى، فإنه من الممكن أن تستخدم العلامـــــات الفارغـــة كاختصارات، وبدلاً من وضع علامة بداية وعلامة نهاية يمكن تضمين علامة واحـــدة فارغـــة. وتتميز العلامات الفارغة عن علامات البداية بالعلامة /> بدلاً من >. فمثلاً بـــدلاً مـــن كتابـــة <PLAYER></PLAYER> يتم كتابة <PLAYER>.

وقد تحتوي العلامات الفارغة على سمات، وهذه هي العلامة الفارغة الخاصة بجو جـيراردي مع العديد من السمات:

```
<PLAYER GIVEN_NAME="Joe" SURNAME="Girardi"
        GAMES="78" AT_BATS="254" RUNS="31" HITS="70"
        DOUBLES="11" TRIPLES="4" HOME_RUNS="3"
        RUNS_BATTED_IN="31" WALKS="14" STRUCK_OUT="38"
        STOLEN_BASES="2" CAUGHT_STEALING="4"
        SACRIFICE_FLY="1" SACRIFICE_HIT="8"
        HIT_BY_PITCH="2"/>
```

وتتعامل محللات XML مع هذه العلامات بنفس الطريقة التي تتعامل بها مع العلامات غـــير الفارغة.وهذا العنصر PLAYER مساوي تماماً "ولكن ليس مطابقاً" لعنصر PLAYER السابق الذي تم تكوينه مع العلامة الفارغة.

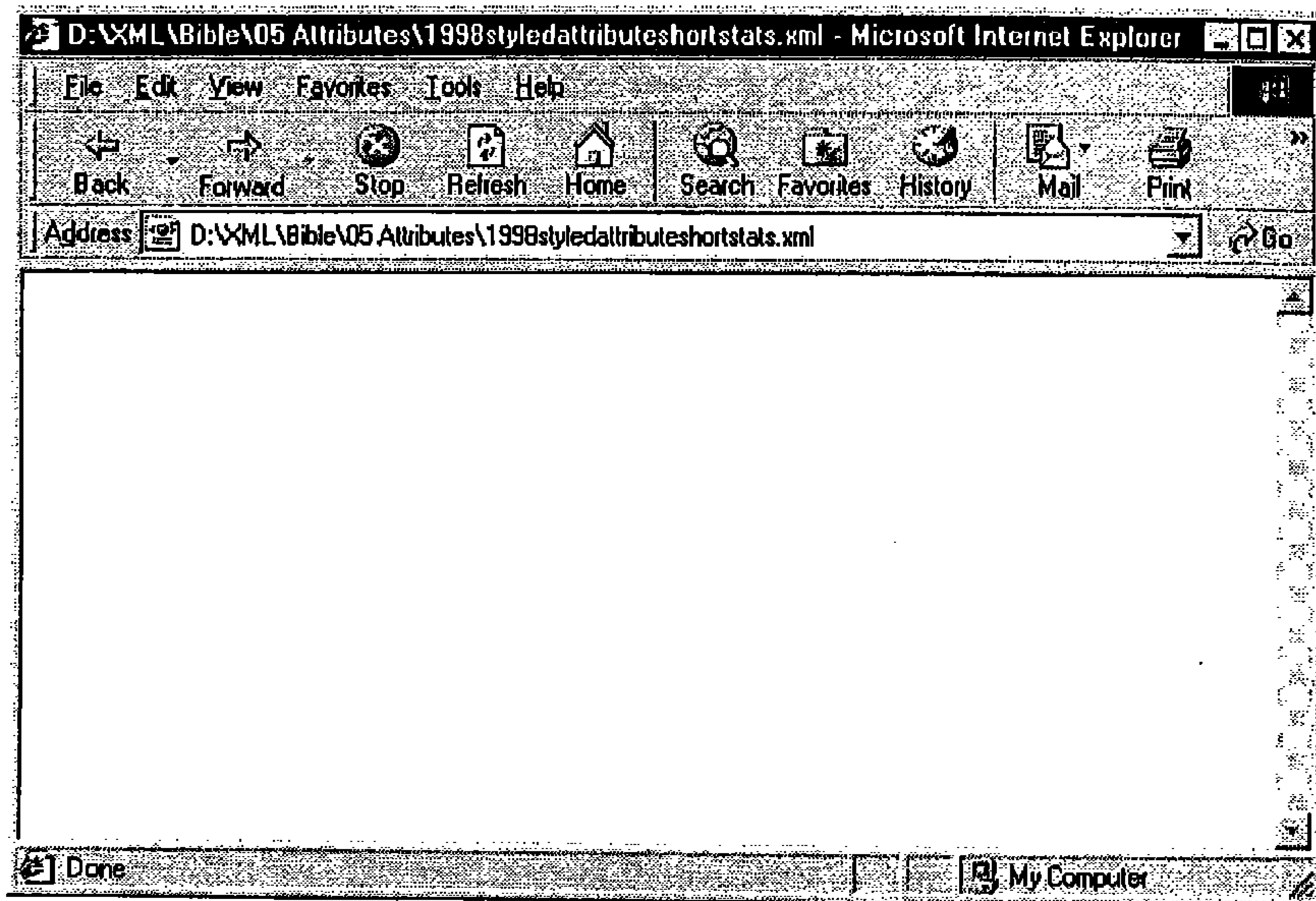
```
<PLAYER GIVEN_NAME="Joe" SURNAME="Girardi"
        GAMES="78" AT_BATS="254" RUNS="31" HITS="70"
        DOUBLES="11" TRIPLES="4" HOME_RUNS="3"
        RUNS_BATTED_IN="31" WALKS="14" STRUCK_OUT="38"
        STOLEN_BASES="2" CAUGHT_STEALING="4"
```

```
SACRIFICE_FLY="1" SACRIFICE_HIT="8"
HIT_BY_PITCH="2"></PLAYER>
```

والفرق بين <PLAYER> و</PLAYER> ما هو إلا اختلاف في الصيغة. وإذا كنت لا تحب الصيغة الخاصة بالعلامة الفارغة، أو كنت تجد بعض الصعوبة في قراءتها، فليس هناك ضرورة من استخدامها.

XSL

تكون السمات مرئية في عرض مصدر XML الخاص بالمستند كما هو موضح في الشكل ٥-١. غير أنه عند تطبيق ورقة نمط CSS فإن السمات تختفي. ويوضح الشكل ٥-٣ تعليمات برمجة ٥-١ بمجرد تطبيق إحصاءات ورقة نمط البيسبول من الفصل السابق الذي يبدو كمستند فارغ لأن أنماط CSS يتم تطبيقها على محتوى العنصر، وليس على السمات. وإذا استخدمت CSS فمن الأفضل جعل أية بيانات تريد عرضها للقارئ جزءاً من محتوى العنصر بدلاً من سماته.



الشكل ٥-٣ يتم عرض مستند فارغ عند تطبيق CSS على مستند XML والذي لا تحتوي عناصره على أية بيانات.

غير أنه هناك لغة ورقة نمط بديلة تسمح لك بالوصول إلى بيانات السمات كما تسمح لك بعرضها، وهي Extensible Style Language والتي يرمز لها بالرمز (XSL) كما أنه يتم تدعيمها بالبرنامج Internet Explorer 5.0 إلى حد ما. ويتم تقسيم XSL إلى جزئين: التحويلات والتنسيق.

ويمكنك جزء التحويلات الخاص بلغة XSL من استبدال إحدى العلامات بعلامة أخرى. كما يمكن تحديد القواعد التي تقوم باستبدال علامات XML بعلامات HTML القياسية، أو بعلامات HTML مع سمات CSS. كما يمكن أيضاً تنفيذ العديد من المهام بما في ذلك إعادة ترتيب العناصر في المستند وإضافة محتويات إضافية لم تكن موجودة من قبل في مستند XML.

بينما يقوم الجزء الخاص بالتنسيق في XSL بتحديد عرض متميز للمستند على شكل صفحة. ويمكنك تنسيق XSL من تحديد شكل الصفحات بما في ذلك الأعمدة المتعددة، والنص المحيط بالرسوم والمسافات بين السطور وسمة الخط وشكله وما إلى ذلك.فهي مصممة لتكون بقوة تسمح لها بمعالجة المهام الخاصة بتنسيق الصفحة للصفحة وتقوم بالطباعة من نفس مصدر المستند. فعلى سبيل المثال يسمح تنسيق XSL لأحد مستندات XML الذي يحتوي على بعض الإعلانات بتكوين إصدارات الطباعة و الإصدارات المباشرة الخاصة بإحدى الصحف التلفزيونية.غير أن Internet Explorer 5.0 لا يدعم تنسيق XSL. لذلك يتم التركيز على تحويلات XSL في هذا الجزء.


يتم مناقشة تنسيق XSL في الفصل ١٥ "كائنات تنسيق XML".

قوالب ورقة عمل XSL

تحتوي ورقة عمل XSL على قوالب يتم إدخال البيانات من مستند XML إليها.وقد تبدو القالب على هذا النحو.

```
<HTML>
  <HEAD>
    <TITLE>
      XSL Instructions to get the title
    </TITLE>
  </HEAD>
  <H1>XSL Instructions to get the title</H1>
  <BODY>
    XSL Instructions to get the statistics
  </BODY>
</HTML>
```

والأجزاء الماثلة يتم إبدالها بعناصر معينة من XML التي تستطيع نسخ البيانات من مستند XML إلى هذا القالب. ويمكن تطبيق هذا القالب على عدة مجموعات من البيانات. فعلى سبيل المثال إذا كان القالب قد تم تصميمه للعمل على المثال الخاص بلعبة البيسبول، فإنه يتم عرض الإحصاءات من المواسم على نفس ورقة النمط.

وقد يذكرك هذا بجانب الخادم بما في ذلك نظم HTML. وفي الواقع فإنه يشبه تماماً ما يتضمنه جانب الخادم.غير أن التحويلات الأساسية الخاصة بمصدر مستند XML وورقة نمط XSL تحدث على العميل وليس على الخادم.وبالإضافة إلى ذلك فإنه لا يلزم أن تكون مخرجات المستند HTML، فمن الممكن أن تكون XML جيدة التكوين.

وتستطيع إرشادات XSL استعادة أية بيانات مخزنة في العناصر الخاصة بمستندXML. ويتضمن ذلك محتوى العنصر وأسماؤه وسماته وهي أهم الأشياء المتعلقة بالمثال المذكور. ويتم اختيار عناصر معينة من أحد النماذج الذي يذكر اسم العنصر وقيمته وأسماء وقيم السمات ومواضعها النهائية والنسبية في شجرة بنية مستند XML وما إلى ذلك. وبمجرد أن يتم اقتباس البيانات من العنصر، فإنه يمكن نسخها وتحريكها وتشكيلها بعدة طرق. ولن يتم هنا عرض كل ما يتعلق بلغة XML، غير أنه سوف يتم شرح كيفية استخدام XSL لكتابة بعض المستندات المتميزة التي يمكن أن يتم عرضها في الويب مباشرةً.

المرجع يغطي الفصل ١٤ "لغة تحويل XSL" تحويلات XSL بالتفصيل

جسم المستند

دعنا نبدأ باستخدام أحد الأمثلة البسيطة مع تطبيقها على مستند XML ، وهو مثال إحصاءات البيسبول المعروضة في تعليمات برمجة ٥-١. وتعليمات البرمجة ٥-٢ ما هي إلا ورقة نمط XSL. وهي تشبه ملف لغة HTML المتضمنة داخل عنصر XSL: Template وبعبارة أخرى تبدو بنيتها هكذا:

تعليمات برمجة ٥-٢: ورقة نمط XSL.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
```

```
<HTML>
  <HEAD>
    <TITLE>
     Major League Baseball Statistics
    </TITLE>
  </HEAD>
  <BODY>
    <H1>Major League Baseball Statistics</H1>
    <HR></HR>
    Copyright 1999
    <A HREF="http://www.macfaq.com/personal.html">
     Elliotte Rusty Harold
    </A>
<BR />
    <A HREF="mailto:elharo@metalab.unc.edu">
     elharo@metalab.unc.edu
    </A>
  </BODY>
 </HTML>
 </xsl:template>
</xsl:stylesheet>
```

وهو يشبه ملف HTML المتضمن داخل عنصر xsl: template، وبعبارة أخرى تبدو بنيتـــها على هذا النحو:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
  HTML file goes here
 </xsl:template>
</xsl:stylesheet>
```

ولا تعد تعليمات البرمجة ورقة نمط في XSL فقط وإنما تعد أيضا مستنداً جيد التنسيق للغـــة XML. وهي تبدأ بإعلان للغة XML. والعنصر الجذري لهذا المستند هـــو XSL: Stylesheet. وتحتوي ورقة النمط هذه على قالب واحد فقط لبيانات لغة XML التي تم وضع تعليماتها البرمجية على إنها عنصر XSL: Template ويوجد لعنصر XSL: Template سمة Match وقيمة / كما

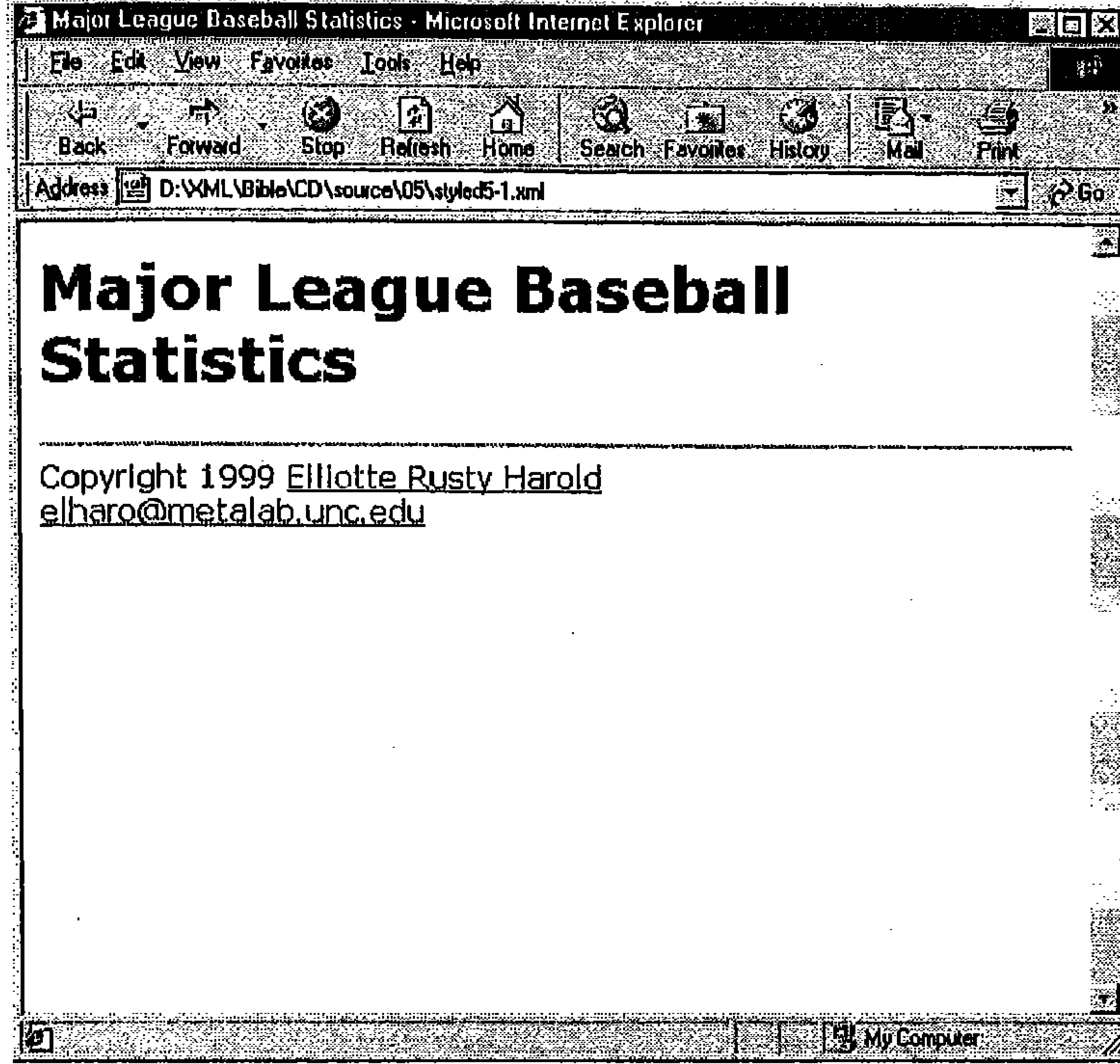
يعد محتواها مستنداً جيد التنسيق للغة HTML. وليست مصادفة أن يكون إخراج لغة HTML جيد التنسيق. وبما أن لغة HTML جزءاً من ورقة نمط XSL ولان ورقة نمط XSL مستند جيد التنسيق للغة XML فلابد على لغة HTML في ورقة نمط XSL أن تكون جيدة التنسيق.

ويحاول مستعرض ويب ربط أجزاء مستند لغة XML في مقابل كل عنصر XSL: Style Sheet. ويقوم القالب/يربط جذر المستند أي المستند بأكمله. ثم يقرأ المستعرض القالب ويدخل البيانات من مستند XML حيث أشارت تعليمات XSL. وعلى أية حال لا يحتوي هذا القالب على أية تعليمات تخص XSL، لذا تعد محتوياته أحرف تم نسخها داخل مستعرض ويب والتي تسفر عن الإخراجات الموضحة في شكل ٥-٤. لاحظ أن شكل ٥-٤ لا يعرض أي بيانات من مستند XML ولكن فقط من قالب XSL.

وإلحاق ورقة نمط XSL للتعليمات البرمجية ٥-٢ بمستند لغة XML في تعليمات البرمجة ٥-١ يكون بطريقة مباشرة. فبكل بساطة يتم إضافة تعليمات البرمجة المعالجة <?xml-style sheet?> بسمتها Type وقيمتها Text/ XSL مع سمة href التي تشير إلى ورقة النمط بين إعلان لغة XML وعنصر الجذر. ومثال ذلك:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="5-2.xsl"?>
<SEASON YEAR="1998">
```

وهي نفس طريقة إلحاق ورقة نمط CSS بمستند ما. والاختلاف الوحيد هو أن سمة Type تكون Text/ XSL بدلا من Text/ CSS.



الشكل ٥-٤ تمحى البيانات من على مستند XML، وليس قـــالب XSL، بعد تطبيق ورقة نمط XSL في تعليمات البرمجة ٥-٢.

العنوان

بكل وضوح هناك شيء غير متواجد في الشكل ٥-٤، وهو البيانات! وعلى الرغم من عـــرض ورقة النمط في تعليمات البرمجة ٥-٢ بعض الأشياء "وهو غير ورقة نمط CSS للشـــكل ٥-٣" إلا أنها لا تعرض أي بيانات من مستند لغة XML. ولإضافة هذا يجب استخدام عناصر تعليمـــات XSL لنسخ البيانات من المستند المصدر للغة XML في قالب XSL. وتضيف تعليمـــات البرمجـــة ٥-٣ تعليمات XSL الضرورية لاستخراج سمة YEAR من عنصـــر SEASON وإدخالـــها فـــي TITLE ورأس الصفحة H1 في المستند النهائي. ويوضح شكل ٥-٥ المستند الحالي.

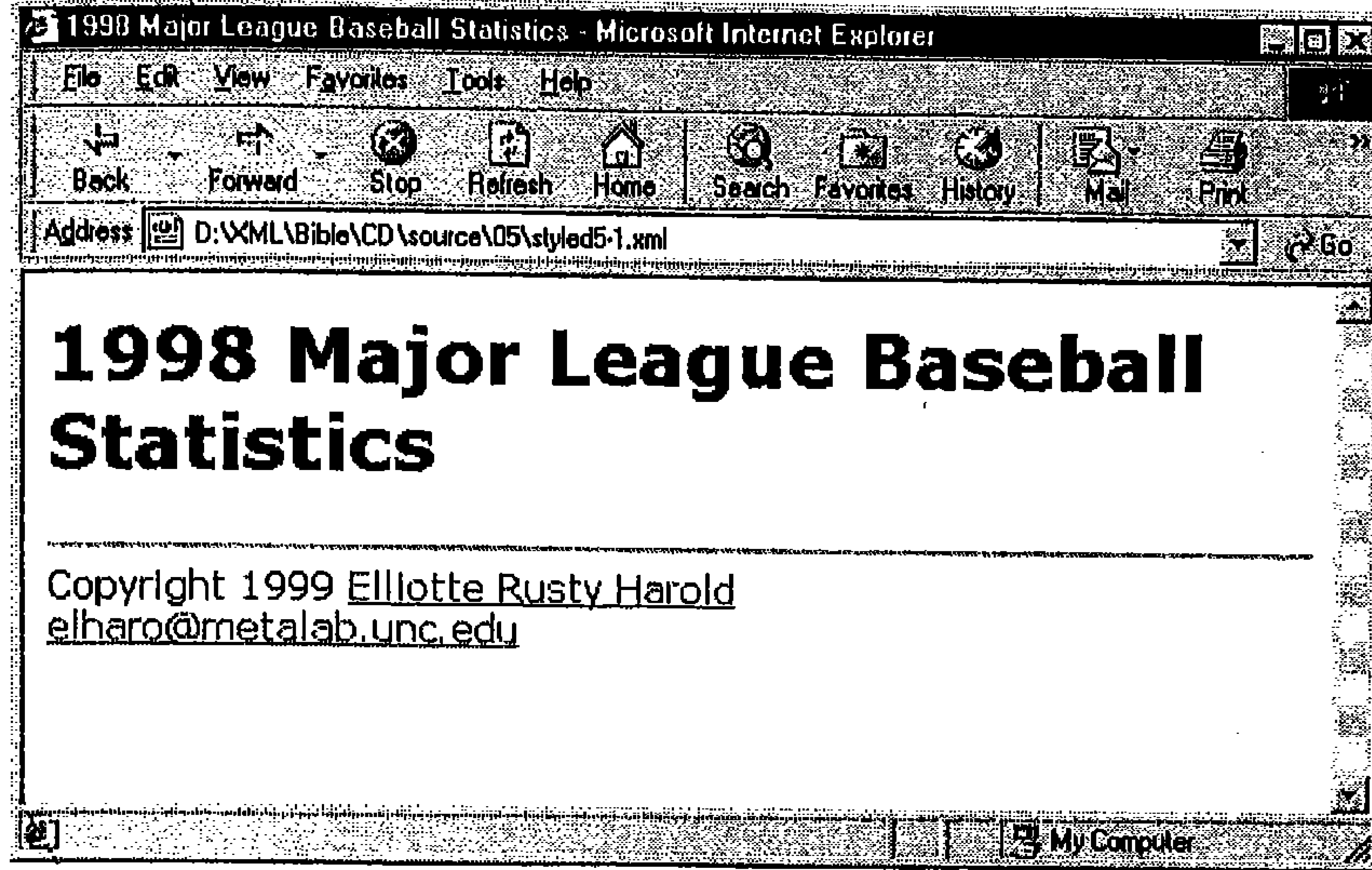
تعليمات البرمجة ٥-٣: ورقة نمط XSL وتعليماتها البرمجية لاستخراج عنصر SEASON وسمة YEAR

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
```

```
<HTML>
  <HEAD>
    <TITLE>
      <xsl:for-each select="SEASON">
        <xsl:value-of select="@YEAR"/>
      </xsl:for-each>
      Major League Baseball Statistics
    </TITLE>
  </HEAD>
  <BODY>
  <xsl:for-each select="SEASON">
    <H1>
      <xsl:value-of select="@YEAR"/>
      Major League Baseball Statistics
    </H1>
  </xsl:for-each>
  <HR></HR>
  Copyright 1999
  <A HREF="http://www.macfaq.com/personal.html">
   Elliotte Rusty Harold
  </A>
  <BR />
  <A HREF="mailto:elharo@metalab.unc.edu">
   elharo@metalab.unc.edu
  </A>
  </BODY>
 </HTML>
</xsl:template>
</xsl:stylesheet>
```

وتعليمات XSL الجديدة لاستخراج سمة YEAR وعنصر SEASON هي:

```
<xsl:for-each select="SEASON">
    <xsl:value-of select="@YEAR"/>
  </xsl:for-each>
```



الشكل ٥-٥ تعليمات البرمجة ٥-١ بعد تطبيق ورقة نمط XSL في تعليمات البرمجة ٥-٣.

وتظهر هذه التعليمات مرتين للرغبة في إظهار السنة مرتين في مستند الإخراج، مرة في رأس الصفحة H1 ومرة في TITLE. وفي كل مرة تظهر، تقوم التعليمات بعمل نفس الشيء كما تبحث تعليمات البرمجة <xsl: for-each Select = "SEASON">. عن كل عناصر SEASON. وتدخل تعليمات البرمجة <xsl: value-of Select = "@YEAR"/> قيمة YEAR لعنصر SEASON أي السلسلة "1998". والذي قام بإيجادها = <xsl: for-each Select "SEASON">.

وبمعنى أخر يبحث xsl: for-each عن عنصر خاص للغة XML في المستند المصدر "تعليمات برمجية ٥-١ في هذه الحالة" والذي تقرأ من خلاله البيانات. وينسخ xsl: value-of جزءاً معيناً للعنصر داخل مستند الإخراج. ويحتاج المستخدم إلى جزئين من تعليمات البرمجة لبرنامج XSL معا لأن جزءاً واحداً من تعليمات البرمجة غير كاف.

ويتم التمييز بين تعليمات XSL وبين عناصر الإخراج مثل HTML وH1 لأن التعليمات توضع في حيز الاسم عن طريق السمة xml ns: xsl للعنصر الجذري لورقة النمط. وفي تعليمات البرمجة ٥-٢ و٥-٣ وجميع الأمثلة الأخرى في هذا الكتاب تكون قيمة هذه السمة:

http://www.w3.org/TR/WD-xsl.


يتم تغطية حيز الاسم بالتفصيل في الفصل ١٨ "Namespaces".

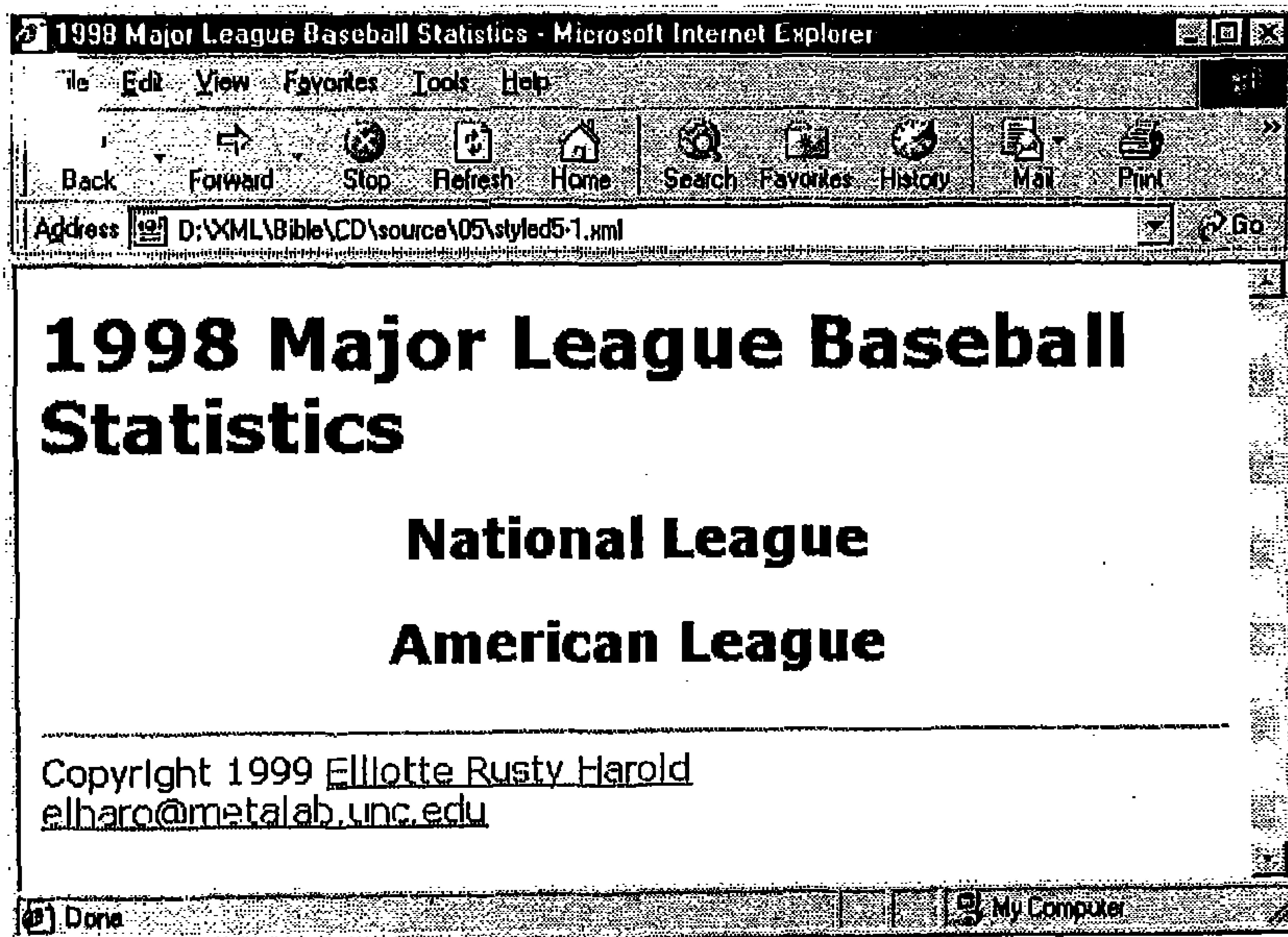
البطولات والتقسيمات والفرق

الخطوة التالية هي إضافة بعض تعليمات XSL لسحب عنصري LEAGUE. وسيتم تفصيل ذلك إلى رأس الصفحات H2. وتوضح تعليمات البرمجة ٥-٤ ذلك.كما يعرض شكل ٥-٦ المستند الحالي مع ورقة النمط هذه.

تعليمات البرمجة ٥-٤: ورقة نمط XSL وتعليماتها البرمجية لاستخراج عناصر LEAGUE

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
  <HTML>
   <HEAD>
    <TITLE>
     <xsl:for-each select="SEASON">
      <xsl:value-of select="@YEAR"/>
     </xsl:for-each>
     Major League Baseball Statistics
    </TITLE>
   </HEAD>
   <BODY>
   <xsl:for-each select="SEASON">
    <H1>
     <xsl:value-of select="@YEAR"/>
     Major League Baseball Statistics
    </H1>
    <xsl:for-each select="LEAGUE">
     <H2 ALIGN="CENTER">
      <xsl:value-of select="@NAME"/>
     </H2>
```

```
    </xsl:for-each>
   </xsl:for-each>
   <HR></HR>
   Copyright 1999
   <A HREF="http://www.macfaq.com/personal.html">
    Elliotte Rusty Harold
   </A>
   <BR />
   <A HREF="mailto:elharo@metalab.unc.edu">
    elharo@metalab.unc.edu
   </A>
   </BODY>
  </HTML>
 </xsl:template>
</xsl:stylesheet>
```



الشكل ٥-٦ عرض أسماء البطولات كرأس صفحات H2 بعد تطبيق ورقة نمط XSL في تعليمات البرمجة ٥-٤.

والمواد الجديدة الأساسية هي التعليمات المبنية xsl: for-each

```
<xsl:for-each select="SEASON">
    <H1>
      <xsl:value-of select="@YEAR"/>
      Major League Baseball Statistics
    </H1>
    <xsl:for-each select="LEAGUE">
      <H2 ALIGN="CENTER">
        <xsl:value-of select="@NAME"/>
      </H2>
    </xsl:for-each>
  </xsl:for-each>
```

وتشير معظم التعليمات إلى اختيار عنصر SEASON. وبعد الاختيار يتم البحث علي سـمة YEAR لهذا العنصر ووضعه داخل <H1> و</H1> مع النص الإضـافي Major League Baseball Statistics. ثم يحلق المستعرض في كل تابع LEAGUE نحـو عنصـر SEASON المختار ويضع قيمة سمة NAME الخاص به بيـن <H2 ALIGN="CENTER"> و</H2>. وبالرغم من وجود واحدة فقط من تعليمات البرمجة xsl:for-each لربط عنصـر JEAGUE إلا أنها تحلق على كل عناصر LEAGUE وهي توابع مباشرة لعنصر SEASON. وبهذا يعمل هـذا القالب في أي مكان بدءاً من الصفر وحتى عدد لا نهائي من البطولات.

وتستخدم نفس التقنية في تعيين رأس الصفحات H3 مع التقسيمات ورأس الصفحات H4 مـع الفرق. وتوضح تعليمات البرمجة ٥-٥ هذا الإجراء كما يوضح الشكل ٥-٧ المستند الحالي مـع ورقة النمط هذه. وتقرأ أسماء التقسيمات والفرق من بيانات لغة XML.

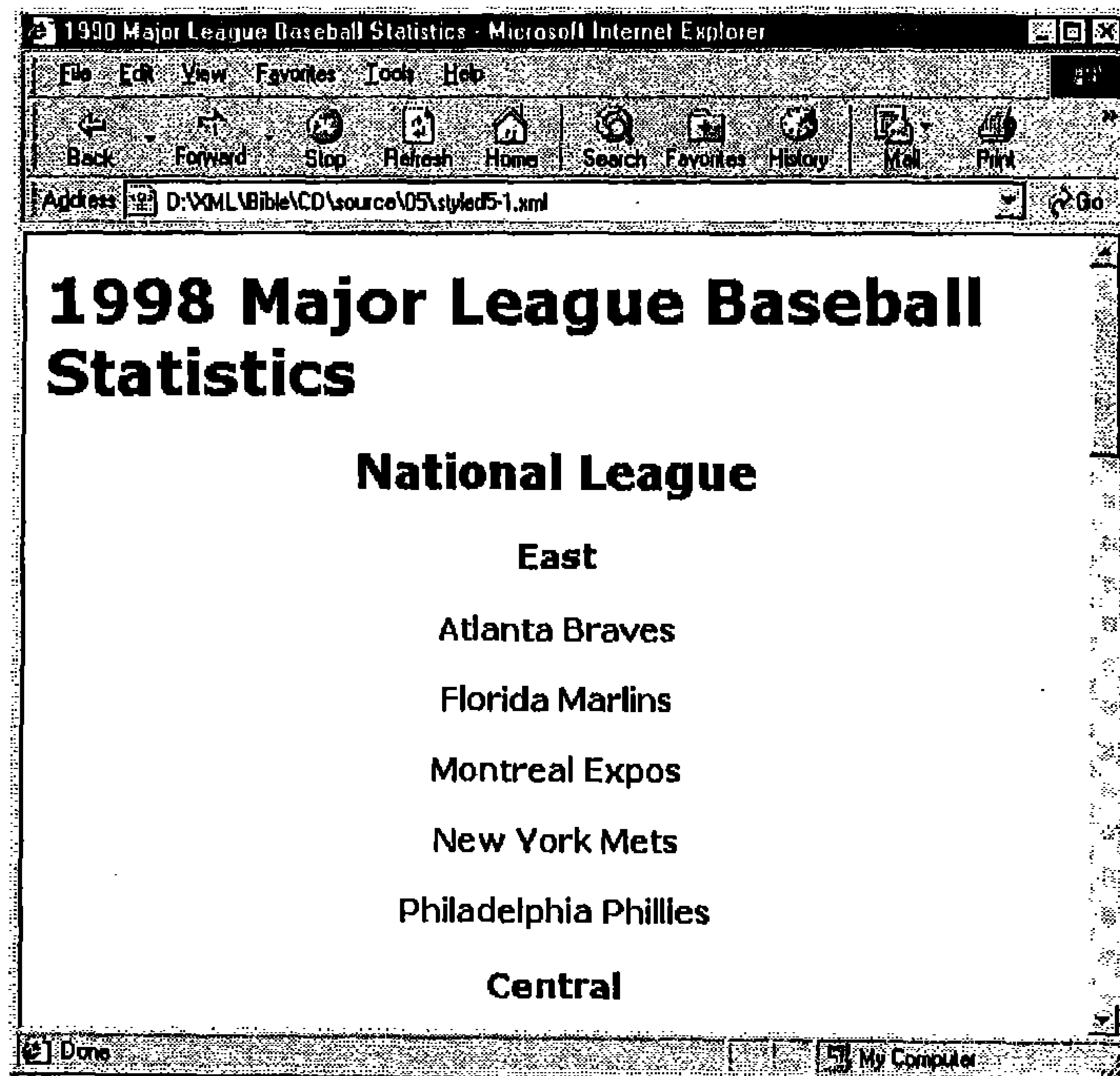
تعليمات البرمجة ٥-٥: ورقة نمط XSL وتعليماتها البرمجية لاستخراج عناصر TEAMو DIVISION

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <TITLE>
          <xsl:for-each select="SEASON">
```

```
    <xsl:value-of select="@YEAR"/>
   </xsl:for-each>
   Major League Baseball Statistics
  </TITLE>
 </HEAD>
 <BODY>

 <xsl:for-each select="SEASON">
  <H1>
   <xsl:value-of select="@YEAR"/>
   Major League Baseball Statistics
  </H1>
  <xsl:for-each select="LEAGUE">
   <H2 ALIGN="CENTER">
     <xsl:value-of select="@NAME"/>
   </H2>
   <xsl:for-each select="DIVISION">
     <H3 ALIGN="CENTER">
     <xsl:value-of select="@NAME"/>
     </H3>
     <xsl:for-each select="TEAM">
       <H4 ALIGN="CENTER">
       <xsl:value-of select="@CITY"/>
       <xsl:value-of select="@NAME"/>
       </H4>
     </xsl:for-each>
   </xsl:for-each>
  </xsl:for-each>
 </xsl:for-each>
 <HR></HR>
 Copyright 1999
 <A HREF="http://www.macfaq.com/personal.html">
  Elliotte Rusty Harold
 </A>
```

```
        <BR />
        <A HREF="mailto:elharo@metalab.unc.edu">
         elharo@metalab.unc.edu
        </A>
        </BODY>
      </HTML>
    </xsl:template>
  </xsl:stylesheet>
```



الشكل ٥-٧ عرض التقسيمات وأسماء الفرق بعد تطبيـــق ورقـــة النمط XSL في تعليمات البرمجة ٥-٥.

في عناصر TEAM، تستخدم قيمتا السمتين CITY وNAME الخاصة بها كمحتويـــات لـــرأس الصفحة H4. لاحظ أيضا أن بناء عناصر xsl: for-each والذي يختار المواســـم والتقســـيمات والبطولات والفرق يشبه الشكل الهيكلي للمستند نفسه. وهذا ليس مصادفة. في حيـــن لا تتطلـــب الأنظمة الأخرى أشكال هيكلية رابطة، وهو أبسط المتطلبات، وخاصة للبيانات مركبة البناء مثـــل إحصاءات البيسبول في تعليمات البرمجة ٥-١.

اللاعبون

الخطوة التالية هي إضافة الإحصاءات للاعبي الفرق. وابسط الطرق لعمل هـــذا هـــو الجـــدول. وتعرض تعليمات البرمجة ٥-٦ ورقة نمط XSL والتي تقوم بترتيب اللاعبين وإحصاءاتهم فـــي الجدول. كما أنه لا يتم تقديم عناصر XSL الجديدة وإنما تستخدم نفس العنـــاصر xsl: for-each وxsl: value-of على عنصر PLAYER وسماته. ويعد الإخراج علامات قياسية لجـــدول لغـــة HTML. ويعرض الشكل ٥-٨ النتائج.

تعليمات البرمجة ٥-٦: ورقة نمط XSL التي تضع اللاعبين وإحصاءاتهم داخل الجدول

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
  <HTML>
   <HEAD>
    <TITLE>
     <xsl:for-each select="SEASON">
      <xsl:value-of select="@YEAR"/>
     </xsl:for-each>
     Major League Baseball Statistics
    </TITLE>
   </HEAD>
   <BODY>
   <xsl:for-each select="SEASON">
    <H1>
     <xsl:value-of select="@YEAR"/>
     Major League Baseball Statistics
    </H1>
    <xsl:for-each select="LEAGUE">
     <H2 ALIGN="CENTER">
      <xsl:value-of select="@NAME"/>
     </H2>
     <xsl:for-each select="DIVISION">
```

```
<H3 ALIGN="CENTER">
<xsl:value-of select="@NAME"/>
</H3>
<xsl:for-each select="TEAM">
  <H4 ALIGN="CENTER">
  <xsl:value-of select="@CITY"/>
  <xsl:value-of select="@NAME"/>
  </H4>
  <TABLE>
   <THEAD>
    <TR>
     <TH>Player</TH><TH>P</TH><TH>G</TH>
   <TH>GS</TH><TH>AB</TH><TH>R</TH><TH>H</TH>
 <TH>D</TH><TH>T</TH><TH>HR</TH><TH>RBI</TH>
<TH>S</TH><TH>CS</TH><TH>SH</TH><TH>SF</TH>
 <TH>E</TH><TH>BB</TH><TH>SO</TH><TH>HBP</TH>
    </TR>
   </THEAD>
  <TBODY>
   <xsl:for-each select="PLAYER">
    <TR>
     <TD>
      <xsl:value-of select="@GIVEN_NAME"/>
      <xsl:value-of select="@SURNAME"/>
     </TD>
     <TD><xsl:value-of select="@POSITION"/></TD>
     <TD><xsl:value-of select="@GAMES"/></TD>
     <TD>
       <xsl:value-of select="@GAMES_STARTED"/>
     </TD>
     <TD><xsl:value-of select="@AT_BATS"/></TD>
     <TD><xsl:value-of select="@RUNS"/></TD>
     <TD><xsl:value-of select="@HITS"/></TD>
     <TD><xsl:value-of select="@DOUBLES"/></TD>
```

```
<TD><xsl:value-of select="@TRIPLES"/></TD>
<TD><xsl:value-of select="@HOME_RUNS"/></TD>
<TD><xsl:value-of select="@RBI"/></TD>
<TD><xsl:value-of select="@STEALS"/></TD>
<TD>
 <xsl:value-of select="@CAUGHT_STEALING"/>
</TD>
<TD>
 <xsl:value-of select="@SACRIFICE_HITS"/>
</TD>
<TD>
 <xsl:value-of select="@SACRIFICE_FLIES"/>
</TD>
<TD><xsl:value-of select="@ERRORS"/></TD>
<TD><xsl:value-of select="@WALKS"/></TD>
<TD>
 <xsl:value-of select="@STRUCK_OUT"/>
</TD>
<TD>
 <xsl:value-of select="@HIT_BY_PITCH"/>
</TD>
</TR>
</xsl:for-each>
</TBODY>
</TABLE>
</xsl:for-each>
</xsl:for-each>
</xsl:for-each>
</xsl:for-each>
<HR></HR>
Copyright 1999
<A HREF="http://www.macfaq.com/personal.html">
 Elliotte Rusty Harold
</A>
```

```
    <BR />
    <A HREF="mailto:elharo@metalab.unc.edu">
     elharo@metalab.unc.edu
    </A>
    </BODY>
   </HTML>
  </xsl:template>
</xsl:stylesheet>
```

الفصل بين رامي الكرة وضارب الكرة

قد تلاحظ في الشكل ٥-٨ عدم معالجة رامي الكرة كما ينبغي. فطوال هذا الفصل والفصل الرابع تم إعطاء رامي الكرة مجموعة مختلفة تماما من الإحصاءات سواء كانت هذه الإحصاءات مخزنة في محتوى العنصر أو السمات ولهذا يحتاج رامي الكرة إلى جدول منفصل عن باقي اللاعبين. وقبل وضع اللاعب داخل الجدول يجب عليك اختباره على أنه رامي الكرة أم لا. فإذا احتوت سمة POSITION الخاصة به على سلسلة "Pitcher" فاسقط هذا اللاعب. ثم اعكس الإجراءات في جدول أخر يضم فقط رامي الكرة. عناصر PLAYER التي تحتوي سمة POSITION الخاصة بها على سلسلة "Pitcher".

ولعمل هذا، يجب وضع تعليمات برمجية إضافية لعنصر xsl: for-each والتي تختار اللاعبين. حيث لا يقوم المستخدم باختيار جميع اللاعبين إلا من تكون سمات POSITION لهم غير رامي الكرة. وتبدو الصيغة هكذا:

```
<xsl:for-each select= "PLAYER[(@POSITION != 'Pitcher')">
```

ولقيام مستند لغة XML بالتمييز بين رامي الكرة البادئ ورامي الكرة البارز، يجب أن تقوم الإجابة الصحيحة باختيار الحالتين معا:

```
<xsl:for-each select= "PLAYER[(@POSITION != 'Starting Pitcher')
  $and$ (@POSITION != 'Relief Pitcher')]">
```

1998 Major League Baseball Statistics - Microsoft Internet Explorer

Address: D:\XML\Bible\CD\source\05\styled5-1.xml

1998 Major League Baseball Statistics

National League

East

Atlanta Braves

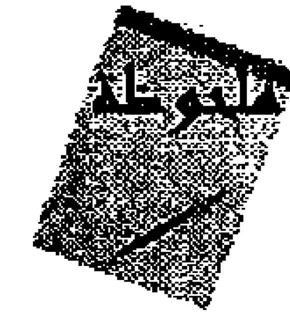
Player	P	G	GS	AB	R	H	D	T	HR	RBI	S	CS	SH	SF	E	BB	SO	HBP
Marty Malloy	Second Base	11	8	28	3	5	1	0	1	1	0	0	0	0	0	2	2	0
Ozzie Guillen	Shortstop	83	59	264	35	73	15	1	1	22	1	4	4	2	6	24	25	1
Danny Bautista	Outfield	82	27	144	17	36	11	0	3	17	1	0	3	2	2	7	21	0
Gerald Williams	Outfield	129	51	266	46	81	18	3	10	44	11	5	2	1	5	17	48	3
Tom Glavine	Starting Pitcher	33	33															
Javier Lopez	Catcher	133	124	489	73	139	21	1	34	106	5	3	1	8	5	30	85	6
Ryan Klesko	Outfield	129	124	427	69	117	29	1	18	70	5	3	0	4	2	56	66	3
Andres	First Base	153	151	555	103	169	27	1	44	121	7	6	0	5	11	63	146	25

Done | My Computer

الشكل ٥-٨ عرض إحصاءات اللاعب بعد تطبيق ورقة نمط XSL في تعليمات البرمجة ٥-٦.

وفي جدول رامي الكرة، يقوم المستخدم منطقيا بعكس هذا إلى الموضع المعادل إما مع "Starting Pitcher" أو "Relif Pitcher" وليس كافيا تغيير not equal إلى equal. فيجب تغيير and إلى or". وتبدو الصيغة هكذا:

```
<xsl:for-each select="PLAYER[(@POSITION = 'Starting Pitcher')
 $or$ (@POSITION = 'Relief Pitcher')]">
```



تستخدم علامة المساواة المفردة فقط لاختبار المساواة بدلا من علامة المساواة المزدوجة المستخدمة في لغة C ولغة Java. وهذا لعدم وجود معادل لعامل التشغيل في XSL.

وتعرض تعليمات البرمجة ٥-٧ ورقة نمط XSL التي تفصل ضارب الكرة ورامي الكرة في جدولين مختلفين. يقوم جدول رامي الكرة بإضافة أعمدة لكل الإحصاءات المعتادة لرامي الكرة. وتقوم تعليمات البرمجة ٥-١ بوضع البرمجة في سمات كالفوز أو الهزيمة أو التعادل الخ. وتستخدم الاختصارات في عناوين الأعمدة لحفظ مرونة نطاق العرض بالنسبة للجدول. يعرض شكل ٥-٩ النتائج.

تعليمات البرمجة ٥-٧: ورقة نمط XSL التي تفصل بين ضارب الكرة ورامي الكرة

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
  <HTML>
   <HEAD>
    <TITLE>
     <xsl:for-each select="SEASON">
      <xsl:value-of select="@YEAR"/>
     </xsl:for-each>
     Major League Baseball Statistics
    </TITLE>
   </HEAD>
   <BODY>
    <xsl:for-each select="SEASON">
     <H1>
      <xsl:value-of select="@YEAR"/>
      Major League Baseball Statistics
     </H1>
     <xsl:for-each select="LEAGUE">
      <H2 ALIGN="CENTER">
       <xsl:value-of select="@NAME"/>
      </H2>
      <xsl:for-each select="DIVISION">
       <H3 ALIGN="CENTER">
       <xsl:value-of select="@NAME"/>
       </H3>
       <xsl:for-each select="TEAM">
        <H4 ALIGN="CENTER">
        <xsl:value-of select="@CITY"/>
        <xsl:value-of select="@NAME"/>
        </H4>
```

```
<TABLE>
 <CAPTION><B>Batters</B></CAPTION>
 <THEAD>
  <TR>
   <TH>Player</TH><TH>P</TH><TH>G</TH>
<TH>GS</TH><TH>AB</TH><TH>R</TH><TH>H</TH>
<TH>D</TH><TH>T</TH><TH>HR</TH><TH>RBI</TH>
<TH>S</TH><TH>CS</TH><TH>SH</TH><TH>SF</TH>
   <TH>E</TH><TH>BB</TH><TH>SO</TH>
   <TH>HBP</TH>
  </TR>
 </THEAD>
 <TBODY>
  <xsl:for-each select="PLAYER[(@POSITION
   != 'Starting Pitcher')
  $and$ (@POSITION != 'Relief Pitcher')]">
   <TR>
    <TD>
     <xsl:value-of select="@GIVEN_NAME"/>
     <xsl:value-of select="@SURNAME"/>
    </TD>
    <TD><xsl:value-of select="@POSITION"/></TD>
    <TD><xsl:value-of select="@GAMES"/></TD>
    <TD>
     <xsl:value-of select="@GAMES_STARTED"/>
    </TD>
    <TD><xsl:value-of select="@AT_BATS"/></TD>
    <TD><xsl:value-of select="@RUNS"/></TD>
    <TD><xsl:value-of select="@HITS"/></TD>
    <TD><xsl:value-of select="@DOUBLES"/></TD>
    <TD><xsl:value-of select="@TRIPLES"/></TD>
    <TD>
     <xsl:value-of select="@HOME_RUNS"/>
    </TD>
```

```
<TD><xsl:value-of select="@RBI"/></TD>
<TD><xsl:value-of select="@STEALS"/></TD>
<TD>
 <xsl:value-of select="@CAUGHT_STEALING"/>
</TD>
<TD>
 <xsl:value-of select="@SACRIFICE_HITS"/>
</TD>
<TD>
 <xsl:value-of select="@SACRIFICE_FLIES"/>
</TD>
<TD><xsl:value-of select="@ERRORS"/></TD>
<TD><xsl:value-of select="@WALKS"/></TD>
<TD>
 <xsl:value-of select="@STRUCK_OUT"/>
</TD>
<TD>
 <xsl:value-of select="@HIT_BY_PITCH"/>
</TD>
</TR>
</xsl:for-each>  <!-- PLAYER -->
</TBODY>
</TABLE>

<TABLE>
 <CAPTION><B>Pitchers</B></CAPTION>
 <THEAD>
  <TR>
   <TH>Player</TH><TH>P</TH><TH>G</TH>
  <TH>GS</TH><TH>W</TH><TH>L</TH><TH>S</TH>
  <TH>CG</TH><TH>SO</TH><TH>ERA</TH>
<TH>IP</TH><TH>HR</TH><TH>R</TH><TH>ER</TH>
<TH>HB</TH><TH>WP</TH><TH>B</TH><TH>BB</TH>
```

```
<TH>K</TH>
</TR>
</THEAD>
<TBODY>
<xsl:for-each select="PLAYER[(@POSITION
 = 'Starting Pitcher')
 $or$ (@POSITION = 'Relief Pitcher')]">
 <TR>
  <TD>
   <xsl:value-of select="@GIVEN_NAME"/>
   <xsl:value-of select="@SURNAME"/>
  </TD>
  <TD><xsl:value-of select="@POSITION"/></TD>
  <TD><xsl:value-of select="@GAMES"/></TD>
  <TD>
   <xsl:value-of select="@GAMES_STARTED"/>
  </TD>
  <TD><xsl:value-of select="@WINS"/></TD>
  <TD><xsl:value-of select="@LOSSES"/></TD>
  <TD><xsl:value-of select="@SAVES"/></TD>
  <TD>
   <xsl:value-of select="@COMPLETE_GAMES"/>
  </TD>
  <TD>
   <xsl:value-of select="@SHUT_OUTS"/>
  </TD>
  <TD><xsl:value-of select="@ERA"/></TD>
  <TD><xsl:value-of select="@INNINGS"/></TD>
  <TD>
  <xsl:value-of select="@HOME_RUNS_AGAINST"/>
  </TD>
  <TD>
   <xsl:value-of select="@RUNS_AGAINST"/>
  </TD>
```




```
      <TD>
       <xsl:value-of select="@EARNED_RUNS"/>
      </TD>
      <TD>
       <xsl:value-of select="@HIT_BATTER"/>
      </TD>
      <TD>
        <xsl:value-of select="@WILD_PITCH"/>
      </TD>
      <TD><xsl:value-of select="@BALK"/></TD>
      <TD>
       <xsl:value-of select="@WALKED_BATTER"/>
      </TD>
      <TD>
      <xsl:value-of select="@STRUCK_OUT_BATTER"/>
      </TD>
     </TR>
    </xsl:for-each>  <!-- PLAYER -->
   </TBODY>
  </TABLE>

 </xsl:for-each> <!-- TEAM -->
 </xsl:for-each> <!-- DIVISION -->
</xsl:for-each> <!-- LEAGUE -->
</xsl:for-each> <!-- SEASON -->

<HR></HR>
Copyright 1999
<A HREF="http://www.macfaq.com/personal.html">
 Elliotte Rusty Harold
</A>
<BR />
```

```
    <A HREF="mailto:elharo@metalab.unc.edu">
     elharo@metalab.unc.edu
    </A>

    </BODY>
  </HTML>
</xsl:template>

</xsl:stylesheet>
```

1998 Major League Baseball Statistics - Microsoft Internet Explorer

Address D:\XML\Bible\CD\source\05\styled5-1.xml

Atlanta Braves

Batters

Player	P	G	GS	AB	R	H	D	T	HR	RBI	S	CS	SH	SF	E	BB	SO	HBP
Marty Malloy	Second Base	11	8	28	3	5	1	0	1	1	0	0	0	0	0	2	2	0
Ozzie Guillen	Shortstop	83	59	264	35	73	15	1	1	22	1	4	4	2	6	24	25	1
Danny Bautista	Outfield	82	27	144	17	36	11	0	3	17	1	0	3	2	2	7	21	0
Gerald Williams	Outfield	129	51	266	46	81	18	3	10	44	11	5	2	1	5	17	48	3
Javier Lopez	Catcher	133	124	489	73	139	21	1	34	106	5	3	1	8	5	30	85	6
Ryan Klesko	Outfield	129	124	427	69	117	29	1	18	70	5	3	0	4	2	56	66	3
Andres Galarraga	First Base	153	151	555	103	169	27	1	44	121	7	6	0	5	11	63	146	25
Wes Helms	Third Base	7	2	13	2	4	1	0	1	2	0	0	0	0	1	0	4	0

Pitchers

Player	P	G	GS	W	L	S	CG	SO	ERA	IP	HR	R	ER	HB	WP	B	BB	K
Tom Glavine	Starting Pitcher	33	33	20	6	0	4	3	2.47	229.1	13	67	63	2		0	74	157

Florida Marlins

Done

الشكل ٥-٩ التمييز بين رامي الكرة واللاعبين الآخرين بعد تطبيق ورقة نمط XSL في تعليمات البرمجة ٥-٧.

محتويات العنصر وسمة الاختيار

في هذا الفصل يتم التركيز على استخدام XSL لتنسيق البيانات المخزنة في سمات العنصر لأنه لا يتم الوصول إليها عن طريق استخدام CSS. وعلى أية حال يعمل XSL بكفاءة عندما يرغب في ضم بيانات حرفية للعنصر بدلا من "أو إضافة إلى" سماتها. وعند ضرورة نسخ نص لعنصر ما

في مستند الإخراج، استخدم ببساطة اسم العنصر كقيمة لسمة Select من عنصر xsl: value-of ومثال ذلك ادرس مرة أخرى تعليمات البرمجة ٥-٨:

```
Listing 5-8greeting.xml<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="greeting.xsl"?>
<GREETING>
Hello XML!
</GREETING>
```

وبافتراض أنك تريد نسخ تحية "Hello XML!" داخل رأس الصفحة H1. استخدم أولا xsl: for-each لاختيار عنصر GREETING:

```
<xsl:for-each select="GREETING">
  <H1>
  </H1>
</xsl:for-each>
```

وهذا يكفي لنسخ علامة H1 في الإخراج. ولوضع نص العنصر GREETING بينهما استخدم xsl: value-of دون سمات select. ثم يتم بشكل افتراضي اختيار محتويات العنصر الحالي (GREETING). وتعرض تعليمات البرمجة ٥-٩ ورقة النمط الكاملة.

تعليمات البرمجة ٥-٩: تحية XSL

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY>
        <xsl:for-each select="GREETING">
          <H1>
            <xsl:value-of/>
          </H1>
        </xsl:for-each>
      </BODY>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```

كما يمكن أيضا استخدم Select لاختيار محتويات عنصر تابع فببساطة اجعل اسم العنصر التابع قيمة لسمة select أي xsl: value-of. ومثال ذلك انظر مثال البيسبول من الفصل السابق والتي تم فيه تخزين إحصاءات كل لاعب في عناصر تابعة بدلا من السمات. وبهذه البنية للمستند "والتي تكون بعيدة الاحتمال عن البنية المعتمدة على السمة في هذا الفصل"، يبدو جدول XSL لضاربي الكرة كما يلي:

```
<TABLE>
          <CAPTION><B>Batters</B></CAPTION>
          <THEAD>
           <TR>
            <TH>Player</TH><TH>P</TH><TH>G</TH>

<TH>GS</TH><TH>AB</TH><TH>R</TH><TH>H</TH>

<TH>D</TH><TH>T</TH><TH>HR</TH><TH>RBI</TH>

<TH>S</TH><TH>CS</TH><TH>SH</TH><TH>SF</TH>

<TH>E</TH><TH>BB</TH><TH>SO</TH><TH>HBP</TH>
            </TR>
          </THEAD>
          <TBODY>
          <xsl:for-each select="PLAYER[(POSITION
           != 'Starting Pitcher')
           $and$ (POSITION != 'Relief Pitcher')]">
           <TR>
            <TD>
             <xsl:value-of select="GIVEN_NAME"/>
             <xsl:value-of select="SURNAME"/>
            </TD>
            <TD><xsl:value-of select="POSITION"/></TD>
            <TD><xsl:value-of select="GAMES"/></TD>
            <TD>
              <xsl:value-of select="GAMES_STARTED"/>
            </TD>
            <TD><xsl:value-of select="AT_BATS"/></TD>
```

```
<TD><xsl:value-of select="RUNS"/></TD>
<TD><xsl:value-of select="HITS"/></TD>
<TD><xsl:value-of select="DOUBLES"/></TD>
<TD><xsl:value-of select="TRIPLES"/></TD>
<TD><xsl:value-of select="HOME_RUNS"/></TD>
<TD><xsl:value-of select="RBI"/></TD>
<TD><xsl:value-of select="STEALS"/></TD>
<TD>
 <xsl:value-of select="CAUGHT_STEALING"/>
</TD>
<TD>
 <xsl:value-of select="SACRIFICE_HITS"/>
</TD>
<TD>
 <xsl:value-of select="SACRIFICE_FLIES"/>
</TD>
<TD><xsl:value-of select="ERRORS"/></TD>
<TD><xsl:value-of select="WALKS"/></TD>
<TD>
 <xsl:value-of select="STRUCK_OUT"/>
</TD>
<TD>
 <xsl:value-of select="HIT_BY_PITCH"/>
</TD>
</TR>
</xsl:for-each>  <!" PLAYER ">
</TBODY>
</TABLE>
```

وفي هذه الحالة، بداخل كل عنصر PLAYER، يتم استخراج محتويات هذا العنصر:

GIVEN_NAME, SURNAME, POSITION, GAMES, GAMES_STARTED, AT_BATS, RUNS,

HITS, DOUBLES, TRIPLES, HOME_RUNS, RBI, STEALS, CAUGHT_STEALING, SACRIFICE_HITS, SACRIFICE_FLIES, ERRORS, WALKS, STRUCK_OUT

وكذلك التابعيين HIT_BY_PITCH ويتم نسخهم في الإخراج. وكما قمنا باستخدام نفس الأسماء للسمات في هذا الفصل كما فعلنا هذا في عناصر التابع PLAYER في الفصل السابق، يماثل هذا المثال المقطع المساوي له في تعليمات البرمجة ٥-٧.والاختلاف الرئيسي بينها هو عدم وجود العلامات @. فهي تشير إلى سمة وليس تابع.

كما يمكنك عمل المزيد عبر سمة Select. فيمكنك اختيار العناصر عن طريق الموضع "على سبيل المثال، الأول، الثاني، الأخير، العنصر السابع عشر، وهكذا". أو بمحتويات معينة، أو قيم سمة معينة، أو تكون للأصليين أو التابعين لها محتويات معينة أو قيم لسمة ما. بل يمكن تطبيق مجموعة كاملة من عامل التشغيل المنطقي Boolean لتجميع شروط الاختيار المختلفة. وسوف نستكشف المزيد من هذه الإمكانات عندما ننتقل إلى XSL في الفصل ١٤.

CSS أو XSL

تتداخل CSS مع XSL إلى حد ما. ولكن بالطبع يعد XSL أكثر قوة من CSS. وعلى اية حال ترتبط قوة XSL ببناءه المعقد. ويتعامل هذا الفصل فقط مع أساسيات XSL. ويعد XSL أكثر تعقيدا وأصعب في الدراسة والاستخدام عن CSS مما يجعلنا نتساءل، "متى يجب علينا استخدام CSS؟ ومتى يجب استخدام XSL؟".

يتم دعم CSS بصورة واسعة عن XSL. فبعض الأجزاء من CSS Level 1 يتم دعمها لعناصر لغة HTML عن طريق Netscape4 وinternet Explorer 4 "بالرغم من وجود اختلافات تثير القلق". وإضافة لذلك هناك احتمال بالدعم الجيد لمعظم CSS Level 1 وبعض من CSS Level 2 وذلك عبر Internet Explorer 5.0 وmozilla5.0 وذلك للغتي XML وHTML. وبهذا فإن اختيار CSS يمنحك المزيد من التوافق مع معدل واسع النطاق من المستعرض.

إضافة إلى ما سبق، يعرف CSS على أنه الأكثر استقرارا. فتعد CSS Level 1 (التي تغطي معظم CSS) وCSS Level 2 توصيات W3C. ولا تزال XSLفي مرحلة الإعداد التي ستأخذ بعض الوقت. وكما وقع مستخدمو XSL الأوائل في الأخطاء، لن يتغير هذا إلا بعد سيادة القياس العام. واختيار CSS يعني أن إعادة كتابة المستخدم لورق النمط من شهر إلى أخر لمتابعة تطورات البرامج والقياس العام قد قل احتماله. وعلى أي حال سيقوم XSL بالعمل على قياس مستخدم.

وبجانب هذا، ولأن XSL حديث للغاية، تقوم البرامج المختلفة بتنفيذ تنويعات ومجموعات جزئية مختلفة للمسودة القياسية. وحتى الآن هناك على الأقل ثلاث تنويعات رئيسية لدى XSL واسعة الاستخدام. ولكن سيتواجد الكثير منها عن قريب وإذا كانت تنفيذات CSS غير الكاملة والشائبة للمستعرضين الحاليين تزعج المستخدم فإن تنوعات XSL ستقوده إلى الجنون.

وعلى أية حال، يعرف XSL بأنه أقوى من CSS. حيث يتيح CSS فقط بتطبيق التنسيق على محتويات العنصر. ولا يتيح تغيير أو إعادة ترتيب هذه المحتويات: كاختيار تنسيق مختلف للعناصر المعتمدة على محتوياتها أو سماتها أو إضافة نص بسيط وإضافي مثل مربع الإمضاء. ويعد XSL أكثر تناسبا عندما تحتوي مستندات XML فقط على أقل البيانات ولا تحتوي على شيء من HTML التي تحيط البيانات.

ومع XSL يمكن فصل البيانات الهامة عن أي شيء أخر من على الصفحة مثل البيانات الإدارية وأشرطة التنقل والإمضاءات. أما مع CSS، يمكن ضم كل هذه الأجزاء في مستندات البيانات. ويتيح XML+XSL مستندات البيانات بالفصل من مستندات صفحة الويب. وهذا يجعل مستندات XML+XSL سهلة الاحتفاظ وأبسط في التعامل معها.

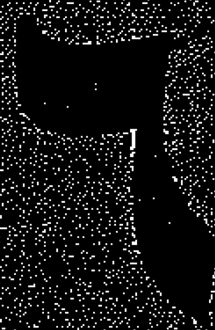
وعلى المدى البعيد يجب أن يصبح XSL الخيار المفضل للتطبيقات مكثفة البيانات وواسعة الانتشار. حيث يتناسب CSS أكثر مع الصفحات البسيطة مثل استخدام عناصر أصل الأصل في إرسال الصور إلى عناصر تابع التابع. ولكن تعد لغة HTML كافية لهذه الاستخدامات. أما إذا وجدت إرهاقا في التعامل مع HTML فلن يفعل XML+CSS هذا. فعلى العكس يأخذك XML+XSL إلى ما هو أبعد من HTML. ولكنك لا تزال في حاجة إلى CSS للتعامل مع المستعرضين القدامى ولكن XSL طويل المدى يبدأ في الانتشار.

خلاصة

في هذا الفصل رأينا أمثلة إنشاء مستند للغة XML من البداية وخاصة:

- تخزين المعلومات في سمة عنصر ما.
- تعد السمة زوج لقيمة الاسم المتضمنة في علامة بداية العنصر.
- تحتفظ السمات بالمعلومات التفصيلية عن العنصر بدلا من بيانات العنصر.
- تتناسب السمات في التعامل معها بصورة أقل من محتويات العنصر.
- تعمل السمات أكثر مع المعلومات البسيطة جدا والتي يستبعد فيها احتمال تغيير شكلها عند تطوير المستند. وعلى الأخص تعمل المعلومات النمطية والرابطة كسمة بطريقة أفضل.
- تعطي العلامات الفارغة جمالا في الصيغة بالنسبة للعناصر التي لا يوجد بها محتوى.
- تعد XSL لغة قوية للنمط والتي تمكنك من الوصول إلى بيانات السمة وعرضها كما تمكنك من تحويل المستندات.

ويتم في الفصل القادم تعيين القواعد الدقيقة والتي يجب التصاق مستندات XML جيدة التنسيق بها. كما سنقوم أيضا باستكشاف وسائل إضافية أخرى لتضمين المعلومات الخاصة بمستندات XML والتي تتضمن التعليقات والتعليمات المعالجة.



الفصل

في هذا الفصل

مستندات XML المُنسقة جيداً

يحتوي HTML 4.0 على مئات العلامات المختلفة ومعظم تلك العلامات لديها الكثير من السمات الممكنة لمتغيرات كثيرة متعددة. ولأن قوة XML تفوق قوة HTML، فذلك يمكن أن يعطيك الإحساس بضرورة معرفة علامات أكثر ولكن ذلك ليس ضرورياً. يحصل HTML على قوته من خلال البساطة والتوسع لا عن طريق العلامات الزائدة عن الحد.

في واقع الأمر لا يعرف XML مسبقاً أية علامات على الإطلاق بل يتيح لك تعريف العلامات الخاصة بك كلما احتجت إليها. وتلك العلامات والمستندات المبنية منهم ليست تحكمية بشكل كامل، بل يجب عليها أن تتبع مجموعة محددة من القواعد التي سيتم شرحها في هذا الفصل. ويدعي المستند الذي تتبع تلك القواعد بمستند Well-Formed والتنسيق الجيد هو المقياس الرئيسي الضروري لمعالجي XML وللمستعرضات لقراءة الملفات. ستعرف في هذا الفصل القواعد اللازمة لمستندات XML Well-Formed واللازمة لـ HTML Well-Formed أيضاً.ويتم لفت النظر بوجه خاص إلى اختلاف XML عن HTML.

ما تتكون منه مستندات XML "لغة ترميز النص المرتبط"

يحتوي مستند XML على مستند يشتمل على ترميز XML وبيانات الحرف وهي مجموعة متتابعة من البايتات محددة الطول والتي تلتزم بقيود معينة. وهي يمكن أن تكون ملف أولاً، فعلى سبيل المثال، مستند XML يمكن أن يكون:

- مخزناً في قاعدة بيانات.
- قد تم إنشاؤه وهو نشط في الذاكرة عن طريق برنامج CGI.
- دمجاً لملفات متعددة تم تضمين كل منهما داخل الآخر.
- غير متواجداً أبداً في ملف خاص به. ومع ذلك لا يضر التفكير في مستند XML كما لو كان ملفاً ما دمت تدرك أنه من الممكن ألا يكون ملفاً في قرص متحرك.

تتكون مستندات XML من وحدات تخزين تدعى كينونات. وتحتوي كل كينونة على نص أو بيانات ثنائية كل على حدا ولا يجتمعان أبداً. وتشتمل بيانات كل نص على أحرف وتستخدم البيانات الثنائية للرسوم والتطبيقات الصغيرة وما إلى ذلك. لسرد نموذج ملموس، ملف HTML الخام الذي يحتوي على علامة <IMG> هو كينونة وليس مستند، بينما ملف HTML مضافاً إليه كل الصور المضمنة فيه بعلامات <IMG> هو مستند كامل.

سأعالج في هذا الفصل والفصول التي تليه مستندات XML البسيطة المكونة من كينونة واحدة وهي المستند نفسه. وستحتوي هذه المستندات بيانات النص لا البيانات الثنائية مثل الصور أو التطبيقات الصغيرة. ومثل تلك المستندات يمكن فهمها دون قراءة ملفات أخرى وهذا يعني أنها قائمة بذاتها.

ويحتوي مثل هذا المستند بطبيعة الحال على سمة Standa Lone في تعريف XML الخاص به مع قيمة Yes كما في المثال التالي:

```
<?xml version="1.0" Standa lone="Yes"?>
```

يمكن استخدام الكينونات الخارجية ومراجع الكينونات لدمج ملفات متعددة ومصادر بيانات أخرى لإنشاء مستند XML واحد. وتلك المستندات لا يمكن توزيعها دون الرجوع إلى ملفات أخرى. وهي تحتوي على سمة Standa lone في إعلان XML ولها قيمة no.

```
<?xml version="1.0" Standa lone="no"?>
```



ستتم مناقشة الكينونات الخارجة ومراجع الكينونات في الفصل التاسع تحت عنوان Entites and External DTD Subsets.

بيانات الحرف والترميز

مستندات XML هي نص في ذاتها ويتكون النص من أحرف والحرف أبجدي أو رقم أو علامة ترقيم أو مسافة أو جدول أو ما شابه ويستخدم XML مجموعة أحرف Unicode والتي لا تحتوي فقط على الأحرف والرموز المعتادة في الأبجدية وباقي لغات أوربا الغربية بل أنها تحتوي أيضاً على الأبجديات الخاصة باللغة السريالية واليونانية والعبرية والديفاناجارية. بالإضافة إلى ذلك، فهي تحتوي على الرموز الأيدغرافية الخاصة Han الشائعة لأبجديات اللغة الصينية واليابانية والمقاطع اللفظية من اللغة الكورية. في هذا الفصل سأقتصر الحديث على النص الإنجليزي.



تتم مناقشة مجموعات الأحرف العالمية في الفصل السابع تحت عنوان:

Foreign Languages and Non-Roman Text.

يؤدي نص مستند XML وظيفتين وهما بيانات الحرف والترميز وبيانات الحرف هي المعلومة الأساسية الخاصة بالمستند، أما الترميز فيقوم بوصف البناء المنطقي للمستند.

على سبيل المثال تذكر Listring 3-2, greeting.xml من الفصل الثالث والمعادة في الصفحة التالية.

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
</GREETING>
```

<CREETING> و <?xml version="1.0" standalone="yes"?>
و</GREETING> هم علامات Makup بينما بيانات الحرف ممثلة في Hello XML ومن المميزات العظيمة التي يتفوق فيها XML على باقي التنسيقات هي أنه يفصل بوضوح البيانات الفعلية الخاصة بالمستند من الترميز الخاص به.

حتى نكون أكثر دقة، يتضمن الترميز كل التعليقات ومراجع الأحرف ومراجع الكينونة ومحددات أقسام DATA، والعلامات وأوامر المعالجة و DTDS. وماعدا ذلك فهو بيانات حرف ولكن هذا خادع لأن عند معالجة المستند يتحول بعض. الترميز إلى بيانات حرف. على سبيل المثال، يتحول الترميز gt & إلى علامة أكبر من (<). تم استبدال بيانات الحرف التي تبقت بعد معالجة المستند وكل الترميز التي تعمل عمل بيانات حرف محدد، ببيانات الحرف الأصلية التي تمثلها وهي تدعي بيانات الحرف الشاملة.

تعليقات

تعليقات XML تماثل تعليقات HTML تماماً فهي تبدأ بـ ١٠٠> وتنتهي بـ < -- ويتم تجاهل كل البيانات بين -! > و< - بواسطة معالج XML كما لو كانت غير موجودة. يمكن استخدام التعليقات لعمل ملاحظات لنفسك أو للتعليق على أجزاء من المستند غير جاهزة على سبيل المثال:

```
<?xml version="1.0" standalone="yes"?>
<!--This is Listing 3-2 from The XML Bible-->
<GREETING>
Hello XML!
<!--Goodbye XML-->
</GREETING>
```

هناك بعض القواعد التي يجب ابتاعها عند استخدام التعليقات وتحدد هذه القواعد كما يلي:

١ - لا تأتي التعليقات قبل تعريف XML وهو أول ما يأتي في المستند على سبيل المثال، المثال التالي غير مقبول:

```
<!--This is Listing 3-2 from The XML Bible-->
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
<!--Goodbye XML-->
</GREETING>
```

٢ - لا توضع التعليقات داخل علامة. فمثلاً المثال التالي غير سليم:

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
```

```
Hello XML!
</GREETING <!--Goodbye--> >
```

٣-يمكن استخدام التعليقات للإحاطة بالعلامات وإخفائها. في المثال التالي، علامة <antigreeting> وكل إتباعها لا يتم التعليق عليهم ولا يظهروا عند إرجاع المستند كما لو أنهم غير موجودين أصلاً

```
<?xml version="1.0" standalone="yes"?>
<DOCUMENT>
<GREETING>
Hello XML!
</GREETING>
<!--
<ANTIGREETING>
Goodbye XML!
</ANTIGREETING>
-->
</DOCUMENT>
```

وحيث أن التعليقات تلغي أجزاء من النص، فيجب توخي الحذر للتأكد من أن النص المتبقي مازال مستند XML و Well-Formed على سبيل المثال، كن حذراً ألا تعلق علامة بادئة إلا إذا قمت بالتعليق على علامة النهاية المقابلة. المثال التالي مثلاً غير صائب:

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
<!--
</GREETING>
-->
```

وبمجرد إزالة النص الذي تم التعليق عليه ما يتبقى هو:

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
```

وبمجرد عدم توافق علامة <GREETING> بعلامة إغلاق </GREETING> لم يعد هذا المستند مستند XML منسق.

٤- يمكن ألا تكون سلسلة الوصلتين (--) موجودة داخل التعليق إلا كجزء من علامة البداية أو علامة النهاية. على سبيل المثال، التعليق التالي غير سليم:

```
<!--The red door--that is, the second one--was left open-->
```

وهذا يعني أنك لا يمكنك تضمين تعليقات كما يلي:

```
<?xml version="1.0" standalone="yes"?>
<DOCUMENT>
  <GREETING>
    Hello XML!
  </GREETING>
<!--
  <ANTIGREETING>
    <!--Goodbye XML!-->
  </ANTIGREETING>
-->
</DOCUMENT>
```

ويعني هذا أيضاً أنك من الممكن أن تواجه مشكلات في أثناء تعليق مصدر شفرة لـــ C أو Java أو Javascript والمليئة بالتعبيرات مثل I أو number left. وبوجه عام تفادي هذه المشكلة ليس بالأمر الصعب ما دمت قد عرفت موضع المشكلة.

مراجع الكينونات

مراجع الكينونات هي علامات تم استبدالها ببيانات الحرف عندما يتم تضمين المستند. ويقوم XML بالتعريف المسبق لمراجع الكينونة الخمسة الموجودة في جدول ٦-١. وتستخدم مراجع الكينونة في مستندات XML محل أحرف محددة سيتم فهمها كجزء من الترميز. على سبيل المثال، يمثل مرجع كينونة &lt علامة (<) والتي يمكن فهمها على أنها بداية لعلامة.

الجدول ٦-١
مراجع كينونة XML المعرفة مسبقاً

مرجع الكينونة	الحرف
&	&
<	<
>	>
"	"
'	'



على خلاف HTML، يجب أن تنتهي مراجع الكينونة في XML بعلامة وقف ; "الفاصلة المنقوطة". لذلك. > مرجع كينونة صحيح بينما &gt ليس كذلك.

دائماً ما يتم فهم علامات أصغر من علامة الضم (&) في نص XML العادي، كعلامات بادئة ومراجع كينونة على التوالي. والنص غير العادي هو مقاطع CDATA المشروح أسفل. لذلك علامات أقل من وعلامات الضم يجب أن يتم تشفيرهم على هيئة > و & على التوالي. فعلى سبيل المثال، يمكنك كتابة عبارة "Ben & Jerry's New York Super Fudge Chunk Ice Cream" كما يلي Ben & Jerry;s New York Super Fudge Chunk Ice Cream. يمكن تشفير علامات أكبر من والاقتباس المزدوج والفواصل الفومية عندما يتم فهمهم كجزء من الترميز.

لذلك فمن السهل تشفير كل منهم بدلاً من محاولة اكتشاف ما إذا كان استخدام محدد يمكن أو لا يمكن فهمه كعلامة ترميز كما يمكن استخدام مراجع الكينونة في قيم السمات مثل

```
<PARAM NAME="joke" VALUE="The diner said,
 &quote;Waiter, There&apos;s a fly in my soup!&quote;">
</PARAM>
```

CDATA

في أغلب الوقت، يكون كل ما في داخل زوجين من أقواس الزوايا (<>) ترميزاً وكل ما هو ليس بالداخل بيانات حرف. لكن هناك استثناء واحد، يكون كل النص في مقاطع CDATA عبارة عن بيانات حرف أصلية. وكل ما يبدو مثل العلامة أو مرجع الكينونة هو نص العلامة أو مرجع الكينونة ولا يحاول معالج XML أن يشرحه بأي طريقة.

وتستخدم مقاطع CDATA عندما تريد أن يتمك فهم النص كبيانات أصلية لا كترميز. وهذا مفيد مبدئياً عندما يكون لديك كتلة كبيرة من النص تحتوي على الكثير من > و< و& أو أحرف ولكنها لا تحتوي على ترميز. وسيكون هذا صحيحاً لكثير من مصادر شفرة C و Java.

تشكل مقاطع CDATA أهمية كبيرة إذا كنت تحاول الكتابة عن XML في XML. فعلى سبيل المثال، يحتوي هذا الكتاب على كتل كثيرة صغيرة من شفرات XML، ومعالج الكلمة الذي استعمله لا يهتم بذلك. لكن إذا كنت سأحول هذا الكتاب إلى، سيكون إلزاماً على استبدال كل علامات أصغر من بعلامة < وكل علامات الضم بعلامة & كما فعلت فيما يلي:

```
&lt;?xml version="1.0" standalone="yes"?&gt;
&lt;GREETING&gt;
Hello XML!
&lt;/GREETING&gt;
```

لتجنب الأضرار لعمل ذلك، يمكن استخدام مقطع CDATA للإشارة إلى أن كتلة نص سيتم عرضها دون ترجمة. وتبدأ مقاطع CDATA بـ <![CDATA[وتنتهي بـ]]> على سبيل المثال:

```
<![CDATA[
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
</GREETING>
 ]]>
```

النص الوحيد الذي لا يسمح به داخل مقطع CDATA هو محدد إغلاق CDATA وهو]]>. ويمكن للتعليقات أن تظهر في مقاطع CDATA ولكنها لا تعمل عمل التعليقات بمعنى أن كلا من علامات التعليق وكل النص الذي تحتوي عليه سيتم إرجاعهم.



بما أن]]> لن تظهر في مقطع CDATA فلا يمكن لمقاطع CDATA أن تضمن. وهذا يجعل من الصعب نسبياً الكتابة عن مقاطع CDATA في XML.

إذا لزم فعل ذلك فعليك استخدام مراجع كينونة و& و < لا تكون الحاجة إلى مقطع CDATA متوافرة لكن إذا حان وقت الحاجة إليها فسيكون الاحتياج إليها شديداً.

العلامات

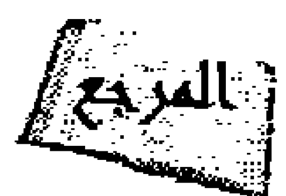
ما يميز ملفات XML عن ملفات النص العادي هو التميز. والجزء الأكبر من الترميز هي العلامات. وإذا كنت قد رأيت في الفصل السابق كيفية استخدام العلامات، فسيقوم الجزء بتعريف ما هي العلامات ويوفر صورة موضحة عن كيفية استخدامها.

باختصار، العلامة هي كل ما في مستند XML وتكون بدايته < ونهايته > . ولا يكون لها نفس شكل علامة HTML. تبدأ علامات البداية بـ < وهي متبوعة باسم العلامة بينما تبدأ علامات النهاية بـ </ وهي متبوعة باسم العلامة أيضاً وأول علامة > تقابلك تقوم بإغلاق العلامة.

أسماء العلامات

لكل علامة اسم ويجب أن تبدأ أسماء العلامة بحرف أو تسطير أسفل السطر (_). يمكن للأحرف التالية في الاسم أن تتضمن حروفاً وأرقام وتسطير أسفل السطر وواصلة وفترات ويمكنهم ألا يتضمنوا مسافات بيضاء. "غالباً ما يحل التسطير أسفل السطر محل المسافات البيضاء". وما يلي بعض علامات XML الصحيحة:

```
<HELP>
<Book>
<volume>
<heading1>
<section.paragraph>
<Mary_Smith>
<_8ball>
```



علامة النقطتين مسموح بها في أسماء العلامات لكنها محجوزة للاستخدام مع مسافات الأسماء. وتساعدك مسافات الأسماء على مزج وتوفيق تعيينات، العلامات التي يمكنها استخدام نفس أسماء العلامات. وتتم مناقشة مسافات الأسماء في الفصل الثامن عشر تحت عنوان Names paces.

فيما يلي علامات XML غير سليمة في بنائها:

```
<Book%7>
<volume control>
<1heading>
<Mary Smith>
<.employee.salary>
```



تطبق قواعد أسماء العلامات على أسماء كثير من الأشياء الأخرى كذلك. وتستخدم القواعد نفسها لأسماء وقيم سمات ID وأسماء الكينونة وعدد آخر من التركيبات التي ستقابلها في الفصول القادمة.

علامات الإغلاق لها نفس الاسم مثل علامات البدء/الفتح لكنها تبدأ بـ / بعد قوس الزاوية البادئ. على سبيل المثال، إذا كانت العلامة البادئة <Foo>، فعلامة الإغلاق تكون </Foo>. وهذه هي علامات النهاية لمجموعة علامات البدء الصحيحة السابقة.

```
</HELP>
</Book>
</volume>
</heading1>
</section.paragraph>
</Mary_Smith>
</_8ball>
```

أسماء XML هي تحسس حالة الأحرف ويختلف هذا عن HTML حيث أن <P> و<p> همـا نفس العلامة بينما </P> يمكنها إغلاق علامة <P>. وما يلي ليس علامات نهايـــة لمجموعـــة علامات البداية الصحيحة التي تناقشها.

```
</help>
</book>
</Volume>
</HEADING1>
</Section.Paragraph>
</MARY_SMITH>
</_8BALL>
```

رغم أن الأحرف الصغيرة والكبيرة يمكن استخدامها في علامات XML ســأعمل فــي هــذا الكتاب على أن تكون العلامات المستخدمة أحرفاً كبيرة لأن ذلك سيجعلها بارزة. لكن في الحالات التي استخدم فيها تعيينات علامة تم تطويرها عن طريق شخص آخر، فسيكون من الضـــروري اختيار حالة الأحرف الاصطلاحية التي يتبعها هذا الشخص.

العلامات الفارغة

الكثير من علامات HTML التي لا تحتوي على بيانات ليس لها علامات إغــلاق. علــى ســبيل المثال، لا توجد علامات </LI> أو </IMG> أو </HR> أو </BR> فــي HTML. يقــوم بعض محررو الصفحات بتضمين علامات </LI> بعد بنود القوائم كمـــا تقــوم بعــض أدوات HTML باستخدام </LI> أيضاً. مع ذلك ينكر مقياس HTML 4.0 الحاجة إلى ذلك.

مثل كل العلامات التي لم يتم التعرف عليها في HTML، تواجد </LI> غير ضرورية ليــس لها تأثير على الإخراج المعالج. لكن ذلك ليس هو الوضع في XML. ما يهتم به XML هو السماح باكتشاف علامات جديدة بينما يتم التعرف على بناء المستند. لذلك لا يمكن تجاهل العلامات غــير المعروفة بهذه السهولة.

يجب على معالج XML أن يكون قادراً على تحديد ما إذا كانت العلامة التي لم يراها من قبــل لها علامة نهاية أم لا.

يقوم XML بالتمييز بين العلامات التي لها إغلاق وتلك التي ليس لها إغلاق العلامات الفارغة بواسطة خط مائل وقوس زاوية مغلق (</) كما في
 أو <HR/>.

يتعامل معالجو الويب الحاليين بشكل متضارب مع مثل تلك العلامات إذا كنت تريد الحصــول على توافق خلفي، يمكنك استخدام علامات الإغلاق بدلاً من ذلك وما عليك سوى عدم تضمين أي نص فيها. على سبيل المثال:

```
<BR></BR>
<HR></HR>
<IMG></IMG>
```

عندما تتعرف على DTDS وأوراق النمط في الفصول القليلة القادمة، ستقابلك طريقتان للحصول على التوافق الأمامي والخلفي الخاص بـ HTML في المستندات التي يجب مراجعتها بواسطة المستعرضات المتوازنة. Legacy browsers.

السمات

كما عرفنا في الفصل السابق، يمكن أن تحتوي علامات البداية والنهاية اختيارياً، على سمات. والسمات هي أزدواج من قيمة الاسم تم فصلها بواسطة علامة يساوي (=) على سبيل المثال،

```
<GREETING LANGUAGE="English">
 Hello XML!
 <MOVIE SRC="WavingHand.mov"/>
</GREETING>
```

وعلامة <GREETING> هنا لها سمة Language والتي لها قيمة English وعلامة <Movie> لها سمة SRC والتي لها قيمة waving Hand move.

أسماء السمات

أسماء السمات هي سلاسل تتبع نفس القواعد من أسماء العلامات لذلك فيجب أن نبدأ أسماء السمات بحرف أو تسطير أسفل السطر (–) ويمكن للأحرف المتتالية في الاسم أن تتضمن أحرفا وأرقاما وتسطيرا أسفل السطر وفواصل وفترات ويمكنهم ألا يتضمنوا مسافات بيضاء لأن الأساطير أسفل السطر يحل محل المسافات البيضاء ويمكن لنفس العلامة ألا تحتوي على سمتين بنفس الاسم فالمثال التالي غير صحيح:

```
<RECTANGLE SIDE="8cm" SIDE="10cm"/>
```

فأسماء السمات حساسة بالنسبة لحالة الأحرف فسمة. Side ليست مثل سمة Side أو سمة Side لذلك فالمثال التالي مقبول:

```
<BOX SIDE="8cm" side="10cm" Side="31cm"/>
```

لكن هذا الآمر مر بك لذلك فمن المستعان عدم كتابة Mar Cup مثل ذلك.

قيم السمات

قيم السمات هي أيضا عبارة عن سلاسل وحتى عندما تعبر السلسلة عن رقم كما في سمة Length المذكورة في المثال التالي فان هذا الرقم هو الحرفان ٧ و٢ لا الرقم العشري 72.

```
<RULE LENGTH="72"/>
```

إذا كنت تكتب شفرة لمعالجة XML ستحتاج ألي تحويل السلسلة إلى رقم قبل القيام بالعمليات الحسابية علية. وعلى عكس أسماء السمات هناك بعض القيود على محتوى قيمة السمة ويمكن لقيم السمات أن تحتوي على مسفات بيضاء أو أن تبدأ برقم أو تحتوى على أي حرف لعلامات الترقيم فيما عدا الاقتباس الفردي أو المزدوج

سمات قيم XML غير محدودة بعلامات اقتباس وعلى عكس سمات HTML فأن قيم سمات XML يجب أن تكون مغلقة في الاقتباس وفي أغلب الأحيان يتم استخدام الاقتباس المزدوج لكن إذا كانت قيمة السمات ذاتها تحتوى على اقتباس مزدوج فيمكن وقتها استخدام الاقتباس الفردي على سبيل المثال:

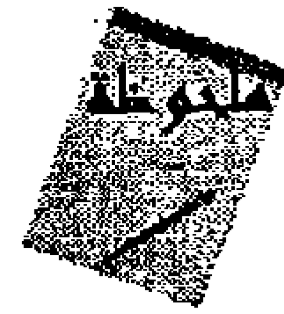
```
<RECTANGLE LENGTH="7" WIDTH="8.5"/>
```

إذا كانت قيم السمات تحتوى على كل من اقتباس فردي أو مزدوج فالذي لا يتم استخدامه لعدم تحديد السلسلة يجب أن يتم استبداله بمراجع الكينونة المناسبة أحيانا أقوم باستبدالهم جميعا مثل:

```
<RECTANGLE LENGTH="8&apos;7&quot;"
WIDTH="10&apos;6&quot;"/>
```

XML المنظم جيدا في المستندات المستقلة بذاتها

بالرغم من أنك تستطيع إنشاء أي عدد من العلامات التي تريدها يجب اتباع بعض القواعد حتى يكون مستند XML منظما بصورة جيدة لأنة إذا لم يكن المستند منظما بصو. جيدة فكل المحاولات لقراءته أو فهمة تبوء بالفشل في واقع الآمر مواصفات XML تمنع شدة XML PARSERS من محاولة إصلاح وفهم المستندات غير المنظمة جيدا والشي الوحيد الذي يمكن السماح به ل المكيف المطابق هو الإبلاغ عن الخطأ ومن الممكن ألا يبذل جهدا لفهم ما أراده المؤلف ومن الممكن ألا يتجاهل الـ MARKWP غير المنظم جيدا والذي يبعث على الضيق كل ما يستطيع عملة هو الإبلاغ عن الخطأ ثم الخروج.



الهدف هنا هو تجنب صراع توافق الشائبة للشائبة والتي قامت بإعاقة HTML والتي جعلت كتابة محللي ومعالجي HTML صعبة ولأن مستعرضات ويب تسمح بتواجد HTML غير المنظمة جيدا مصممو صفحات الويب لا يبذلوا الجهد للتأكد من أن HTM الخاص بهم سليم بل هم يعتمدون على الشوائب في المستعرضات الفردية للحصول على تأثيرات مميزة ولعرض القاعدة الضخمة المحملة من صفحات HTML بصورة جيدة يجب على كل مستعرض ويب جديد أن يدعم كل فارق صغير أو صفة مميزة لكل مستعرضات الويب السابقة سيتجاهل الزبائن كل مستعرض يتمسك بمقياس HTML ولتجنب هذه المشكلة معالجو XML مطالبون بتقبل XML المنظم جيدا فقط لا غير.

وحتى يكون المستند WELL FORMED يجب على كل بيانات أحرف والترميز فـــي مســتند XML أن يلتزموا بالقواعد المذكورة في الأجزاء السابقة وهناك العديد من القواعد الخاصة بكيفيــة ارتباط العلامات وبينات الأحرف ببعضهم البعض وهذه القواعد هي ما يلي:

١- يجب أن نبدأ المستند بإعلان XML.

٢- يجب على العناصر التي تحتوي على بيانات أن يكون لها علاقات بداية ونهاية.

٣- يجب على العناصر التي لا تحتوي على بيانات وتستخدم علامة واحدة أن تنتـــهي بـــ />.

٤- يجب على المستند أن يحتوي بالضبط على عنصر واحد يشتمل جميع العناصر الأخرى.

٥- يمكن للعناصر أن تضمن لكن لا يمكن لها التجاوز.

٦- يجب أن يتم اقتباس قيم السمات.

٧- يمكن استخدام أحرف <و & فقط لابتداء العلامات ومراجع الكينونة على التوالي.

٨- مراجع الكينونة الوحيدة التي تظهر هي & و < و ' و ".

يجب أن يتم تطبيق هذه القواعد الثمانية على المستندات التي بها DTD، وهناك قواعد إضافية للتنظيم الجيد والتي تعرف العلامة بين المستند وبين DTD وستتم مناقشتها في فصـــول لاحقــة. والآن سنلقي نظرة مفصلة على تلك القواعد الخاصة بالمستند دون DTD.

المرجع تتم مناقشة DTD في الجزء الثاني Part II

#1: يجب أن يبدأ المستند بإعلان XML

هذا هو إعلان XML للمستندات المستقلة بذاتها في XML 1.0

```
<?xml version="1.0" standalone="yes"?>
```

إذا كان الإعلان موجوداً على الإطلاق، فيجب أن يكون أول شيء فـــي الملــف وذلــك لأن معالجي XML يقومون بقراءة البايتات العديدة الأولى للملف ثم مقارنتها بشفرات السلسلة المتعـددة <?xml لتحديد أي تعيينات الأحرف تم استعمالها هل هـــي "UTF-8 أم big-endian Unicode أو little-endian Unicode" لا يجب أن يسبق ذلك أي شيء إلا علامة ترتيب بايت غير مرئية بما فيها المسافات البيضاء.

على سبيل المثال، السطر التالي هو طريقة غير مقبولة لبدء ملف XML وذلك بسبب المسافات الزائدة في واجهة السطر:

```
<?xml version="1.0" standalone="yes"?>
```

المرجع

ستتم مناقشة UTF-8 ومتغيرات Unicode في الفصل السابع تحت عنوان: اللغات الأجنبية والنص غير الروماني. يسمح XML بحذف إعلان XML نهائياً ولكن مثل هذه الممارسة غير مستحبة بوجه عام رغم أن لها استخدامات على فترات متباعدة. فعلى سبيل المثال، حذف إعلان XML يساعدك على إنشاء مستند XML منظم جيداً عن طريق مزج مستندات XML جيداً أخرى، وهذه العملية سنناقشها في الفصل التاسع، وهو يسمح أيضاً بكتابة مستندات HTML منظمة جيداً وهذا سيأتي ذكره لاحقاً في هذا الفصل.

#2: استخدم علامات البداية والنهاية في العلامات غير الفارغة

تقوم مستعرضات الويب بالتغاضي عن عدم إغلاق علامة HTML فمثلاً إذا تضمن المستند علامة <B> دون علامة </B> مقابلة سيكون كل المستند بعد علامة <B> بـالخط الأسود العريض، وسيظل المستند معروضاً مع ذلك. لكن يختلف الحال مع XML، فكل علامة بداية يجب أن يتم إغلاقها بعلامة نهاية مقابلة. وإذا لم ينجح المستند في إغلاق العلامة، فإن المستعرض أو المعالج سيبلغ عن وجود خطأ ولن يقوم بعرض محتويات المستند بأي شكل من الأشكال.

#3: قم بإنهاء العلامات الفارغة بواسطة "/>"

لا تحتاج العلامات التي لا تحتوي على بيانات مثل علامات
 و<HR> و<IMG> الموجودة في HTML، إلى علامات إغلاق لكن يجب تعريف علامات XML عن طريق الإغلاق بـ /> بدلاً من مجرد >.

على سبيل المثال، علامات
 و<HR> و<IMG> المساوية لعلامات XML هي على التوالي
 و<HR/> و<IMG/>. تتعامل مستعرضات الويب الحديثة بدون ترابط مع مثل تلك العلامات. إذا كنت تحاول الحصول على توافق خلفي، يمكنك استخدام علامات إغلاق بدلاً من ذلك ولا تضمن أي نص فيهم. على سبيل المثال:

```
<BR></BR>
<HR></HR>
<IMG></IMG>
```

وحتى في هذه الحالة Netscape له بعض المشاكل مع
 و</BR> "فهو يفهم كلاً منهما على أنهما فواصل سطر بدلاً من أن يفهم الأولى فقط" لذلك فمن غير العملي تضمين علامات فارغة منظمة جيداً في:

#4: اجعل عنصراً واحداً يحتوي على كل العناصر الأخرى

يحتوي مستند XML على عنصر جذر يحتوي على كل العناصر الأخرى للمستند وهذا يستدعي عنصر المستند بدلاً من ذلك. وباعتبار أن عنصر جذر المستند غير فارغ وهذا هو الحال دائماً. فيجب ألا يكون محدداً بعلامات بداية ونهاية وهذه العلامات يمكن لكن ليس لزماً عليها أن يكون لها اسم root أو Document. على سبيل المثال، في المثال التالي عنصر الجذر هو GREETING.

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
</GREETING>
```

إعلان XML ليس عنصراً بل أمر معالجة لذلك فلا يجب أن يكون مضمناً داخل عنصر الجذر. وبالتالي فالبيانات التي ليست على شكل عناصر في مستند XML مثل باقي أوامر المعالجة أو التعليقات، dtds لا يجب أن تكون داخل عنصر الجذر، بينما يجب أن تضمن كل العناصر الفعلية ما عدا الجذر نفسه داخل عنصر الجذر.

#5: لا تتجاوز العناصر

في أغلب الأحيان تحتوي العناصر على عناصر أخرى لكن يمكن للعناصر ألا تتجاوز وهذا يعني أنه إذا كان العنصر يحتوي على علامة بادئة لعنصر آخر، فلا بد أن يحتوي على علامة نهاية دون علامة البداية المطابقة. المثال التالي مقبول في XML:

```
<PRE><CODE>n = n + 1;</CODE></PRE>
```

لكن المثال التالي غير مقبول في XML وذلك لأن علامة الإغلاق </PRE> تأتي قبل علامة الإغلاق </CODE>.

```
<PRE><CODE>n = n + 1;</PRE></CODE>
```

تستطيع معظم مستعرضات HTML التعامل مع هذه الحالة بسهولة لكن مستعرضات XML ملزمون في هذه الحالة بالتبليغ عن وجود خطأ في هذا الإنشاء.

ويمكن للعلامات الفارغة الظهور في أي مكان، فعلى سبيل المثال،

```
<PLAYWRIGHTS>Oscar Wilde<HR/>Joe Orton</PLAYWRIGHTS>S
```

اختلاط هذه القاعدة بقاعدة رقم ٤ يبين أن هناك عنصر واحداً بالضبط لكل العناصر غير الجذرية وهو يحتوي على العنصر غير الجذري لكن لا يحتوي على أي عنصر آخر يحتوي على عنصر غير جذري. ويدعي هذا المحتوى الفوري "الأصل" للعنصر غير الجذري ويتم الإشارة

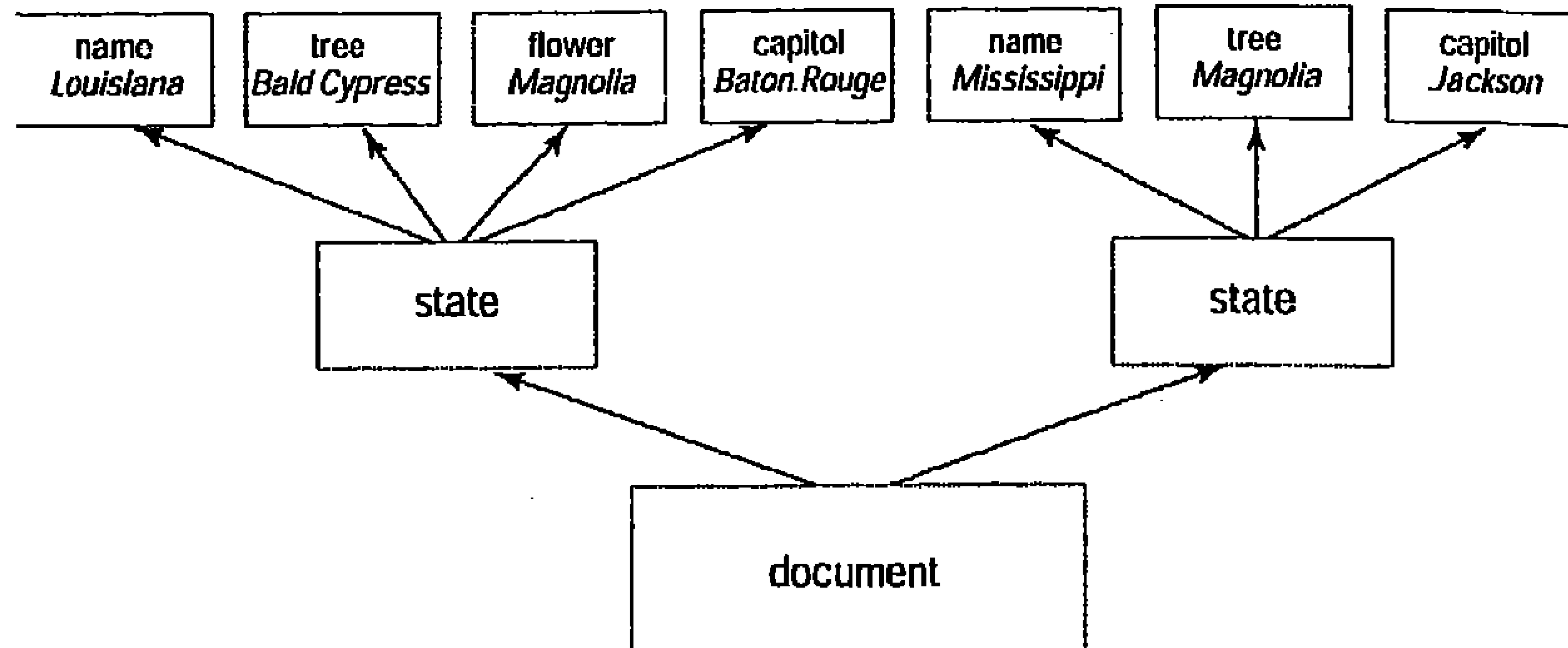
إلى العنصر غير الجذر على أنه تابع لعنصر الأصل. لذلك فكل عنصر غير جذري لـــه أصـــل واحد فقط بينما العنصر الفردي له عدد غير محدود من الاتباع أو حتى ليس لـــه أتباعــاً علـــى الإطلاق.

بالتدقيق في قائمة "٦-١" المبينة أسفل الصفحة فإن عنصر الجذر هو عنصر المســتند وهــو يحتوي على أثنين من حالات الأبناء. عنصر الحالة STATE الأول يحتوي على أربعة أبناء وهم: NAME وTREE وFLOWER وCAPITOL أما عنصر الحالة STATE الثاني فيحتوي على ثلاثة اتباع فقط هم NAME وTREE وCAPITOL. ويحتوي كل تابع منهم على بيانات حرف فقـــط لا على مزيد من الاتباع.

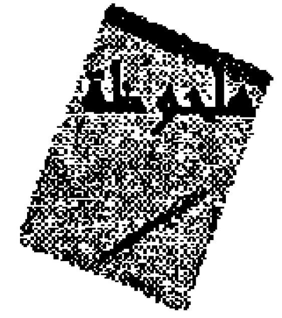
تعليمات البرمجة ٦-١ SEASON يحتوي على تابعي LEAGUE

```
<?xml versi n="1.0" standal ne="yes"?>
<DOCUMENT>
<STATE>
<NAME>L uisiana</NAME>
<TREE>Bald Cypress</TREE>
<FLOWER>Magn lia</FLOWER>
<CAPITOL>Bat n R uge</CAPITOL>
</STATE>
<STATE>
<NAME>Mississippi</NAME>
<TREE>Magn lia</TREE>
<CAPITOL>Jacks n</CAPITOL>
</STATE>
</DOCUMENT>
```

وبلغة المبرمجين هذا يعني أن مستند XML يكون شجرة ويبين شكل ٦-١ شجرة قائمـــة ٦-١ بالإضافة إلى سبب تسمية هذا التشكيل باسم شجرة. فهي تبدأ من الجذر ثم تنطلق إلـــى الأوراق التي في نهاية الشجرة ١. كما أن للشجرة خصائص جيدة تجعل من السهل على برامج الكمبيوتـــر قراءتهم رغم أن ذلك لا يمثل لك أهمية لكونك مؤلف المستند.



الشكل ٦-١ بناء شجرة قائمة "٦-١".

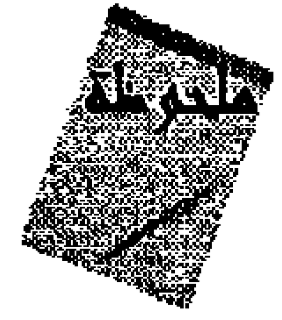


يتم رسم الشجرة من أعلى لأسفل بمعنى أن جذر الشجرة يكون في أعلى الصورة لا في أسفلها. رغم أنها لا تبدو مثل الشجرة الحقيقية إلا أن هذا لا يؤثر على التركيب البنيوي لبناء البيانات على الإطلاق.

#6 قم بتضمين قيم السمات داخل اقتباس

يتطلب XML تضمين كل قيم السمات داخل علامات اقتباس دون الاهتمام ما إذا كانت قيمة السمة تحتوي على مسافات. على سبيل المثال:

```
<A HREF="http://metalab.unc.edu/xml/">
```



هذا ليس صحيحاً في HTML، فعلى سبيل المثال، يسمح HTML للعلامات أن تحتوي على سمات غير مقتبسة.فمثلاً هذه العلامة <A> مقبولة في HTML.

```
<A HREF=http://metalab.unc.edu/xml/>
```

الشرط الوحيد هو أنه لا يجب على قيمة السمة نفسها أن تحتوي على مسافات مضمنة.

إذا كانت قيمة السمة نفسها تحتوي على اقتباس مزدوج، يمكنك استخدام اقتباس فردي للإحاطة بالقيمة بدلاً من ذلك على سبيل المثال:

```
<IMG SRC="sistinechapel.jpg"
ALT='And G d said, "Let there be light,"
and there was light'/>
```

إذا كانت قيمة السمة تحتوي على كل من اقتباس فردي ومزدوج يمكنك استخدام مرجع الكينونة &ap s; للاقتباس الفردي "فاصلة علوية" &qu t; للاقتباس المزدوج. على سبيل المثال:

```
<PARAM name="j ke" value="The diner said,
&qu t;Waiter, There&ap s;s a fly in my s up!&qu t;">
```

#7 لا تستخدم إلا > و& لعلامات البداية والكينونة

يفترض XML أن قوس الزاوية البادئ يبدأ علامة دائماً، وأن علامة الجمع تبدأ مرجـــع كينونـــة "وهذا صحيح غالباً في HTML أيضاً، لكن معظم المستعرضات ستفترض الفاصلة المنقوطـــة إذا ما تركتها" تفكر في هذا المثال:

```
<H1>A H mage t  Ben & Jerry's
New Y rk Super Fudge Chunk Ice Cream</H1>
```

ستقوم مستعرضات ويب بعضها بصورة صحيحة \، لكن لتوخي السلامة الكاملـــة يجـــب أن تتفادى علامة الجمع بـــ & فيكون الحال كما يلي.

```
<H1>A H mage t  Ben &amp; Jerry's New Y rk Super Fudge Chunk
Ice Cream</H1>
```

قوس الزاوية المفتوحة (>) متشابه انظر هذا السطر الشائع من شفرة جافا:

```
<CODE>     f r (int i = 0; i <= args.length; i++ ) { </CODE>
```

يعتبر كل من XML وHTML علامة أصغر من => بداية لعلامة وتستمر العلامة حتى تصل إلى علامة < التالية: لذلك يتم إرجاع السطر كما يلي:

```
f r (int i = 0; i
```

بدلاً من

```
f r (int i = 0; i <= args.length; i++ ) {
```

وتتم ترجمة } (++args.length; i = على أنها جزء من علامة غير معروفة:

يمكن تضمين علامة أصغر من في النص في كل من XML وHTML عن طريق كتابتها على صورة <. مثل:

```
<CODE>     f r (int i = 0; i &lt;= args.length; i++ ) { </CODE>
```

ويتطلب XML المنظم جيداً أن تتم كتابة & على شكل & و> على شكل < كلما تـــم استخدامها لذاتها لا كجزء من علامة أو كينونة.

#8: لا تستخدم سوى مراجع الكينونة الخمسة الموجودين مسبقاً

أنت معتاد على عدد من مراجع الكينونة من HTML فمثلاً &c py; تقوم بإدخال رمز حتى النسخ © أما ® فتدخل رمز الماركة المسجلة ®.

ولكن بالإضافة إلى مراجع الكينونة الخمس التي تمت تعريفها أولاً في DTD وأنت لم تتعرف على DTD بعد. إذا ظهر حرف علامة الجمع & في أي مكان في المستند، يجب أن تليها مباشرة amp; أو lt; أو gt; أو ap s; أو qu t; وكل الاستخدامات الأخرى تتعارض مع التنظيم الجيد.



في الفصل التاسع تحت عنوان المجموعات الفرعية الخاصـــة بالكينونــة وDTD الخارجي، ستعرف كيف تجعل DTD من الممكن تعريف مراجع الكينونة جـــديدة والتي تدخل رموز معينة أو مقدار من النص المتداول.

HTML المنظم الجيد

يمكنك تجربة مهارات XML قبل أن تقوم مستعرضات ويب بدعم XML عن طريق كتابة HTML منظم جيداً. وهذا هو HTML يلتزم بتحديدات XML المنظم جيداً لكنه يستخدم علامـــات HTML القياسية فقط. وHTML المنظم جيداً أسهل في قراءته من HTML غير المتقن الذي يكتبه النـــاس وأدوات WYSIWYG مثل FrontPage كما أنه أسهل في الفهم لأجهزة الويب ومحركات البحث التلقائي وخهو عنيف لا يمكن كسره بسهولة عند إجراء أي تغيير كما أنه ليس معرضاً لاختلافات المستعرضات المتضاربة أو نظم التشغيل المتضاربة عند الفهم. كما يمكنك استخدام أدوات XML للعمل في مستندات HTML مع الحصول على توافق ارتجاعي للقراء الذين لا تقوم مستعرضاتهم بدعم XML.

مشاكل صفحات ويب الحقيقية

صفحات الويب الحقيقية غير متقنة فالعلامات ليست مغلقة والعنـــاصر تتجـــاوز ويتـــم تضميـــن علامات أصغر من الأصلية داخل الصفحات وتم حذف الفواصل المنقوطة من نـــهايات مراجـــع الكينونة. وصفحات الويب التي بها مثل تلك المشاكل هي غـــير صالحـــة لكـــن تقـــوم معظـــم مستعرضات ويب بقبولها. لكن إذا أصلحت هذه المشكلات، ستكون صفحات الويب أكـــثر نقـــاء ويتم عرضها بسرعة أكبر ويسهل الحصول عليها.

بعض المشاكل الشائعة في صفحات الويب هي ما يلي:

١- علامات البداية دون علامات النهاية "عناصر غير مغلقة"

٢- علامات نهاية دون علامات بداية

٣- عناصر متجاوزة

٤- سمات غير مقتبسة

٥- علامات > و< و& و"

٦- عدم تواجد عنصر جذر

٧- حالة أحرف علامة النهاية لا تتناسب مع حالة علامة البداية

لقد سردت المشكلات بأولوية الأهمية وتختلف التفاصيل من علامة لأخرى. فعلى سبيل المثال علامة غير مغلقة مثل<STRONG> ستحول كل العناصر التي تتبعها إلى خط أسود عريض بينما لا تسبب علامات غير مغلقة مثل <LI> أو <P> أية مشاكل على الإطلاق هناك بعض القواعد التي تطبق على مستندات XML فقط، وهذا يمكن أن يسبب مشاكل إذا حاولت دمجها في صفحات HTML الموجودة وهي تتضمن:

١- البدء بإعلان XML

٢- يجب أن يتم إغلاق العلامات الفارغة بواسطة />

٣- مراجع الكينونة الوحيدة المستخدمة هي & و< و> و&ap s; و&qu t;

إصلاح هذه المشكلات ليس صعباً لكن هناك مناطق يجب الحذر عندها وسنكتشفها فيما يلي:

اغلق كل علامات البداية

يجب عن كل عنصر يحتوي على محتوى سواء كان نصاً أو أي عنصر ابن آخر، أن يكون لديه علامة بداية وعلامة نهاية. HTML لا يتطلب ذلك فعلى سبيل المثال يمكن استخدام <P> و<DT> <DD> و<LI> كل على حدا. لكن القيام بذلك يعتمد على قدرة مستعرض الويب على القيام بتخمين جيد بالنسبة للمكان الذي ينتهي فيه العنصر والمستعرضات لا تقوم دائماً بما يطلبه المؤلف أو يتوقعه لذلك فمن الأفضل إغلاق كل علامات البداية.

التغيير الكبير الذي يتطلبه في كيفية كتابة HTML هو التفكير في <P> كمحتوى لا كعلامة فاصلة للفقرة. على سبيل المثال، يمكنك تنسيق بداية الأوراق الفيدرالية كما يلي:

```
T thePe ple f the State f New Yrk:
<P>
AFTER anunequivcal experience fthe inefficiency f the
subsistingfederal gvernment y u are called upn t
deliberate n a new Cnstitutin f r the United States f
America.The subject speakits wn imp rtance c mprehending
in its cnsequencesn thing less than the existence f the
UNION, the safety andelfare fthe parts f which it is
c mp sed the fate f an empire in many respects thestn
interestingin the wrld. It has been frequently remarked that
```

it seems t have been reserved t the pe ple f this c untry,
by their c nduct and example, t decide the imp rtant questi n,
whether s cieties f men are really capable r n t f
establishing g d g vernment fr m reflecti n and ch ice, r
whether they are f rever destined t depend f r their p litical
c nstituti ns n accident and f rce. If there be any truth in
the remark, the crisis at which we are arrived may with
pr priety be regarded as the era in which that decisi n is t
be made; and a wr ng electi n f the part we shall act may, in
this view, deserve t be c nsidered as the general misf rtune
f mankind.
<P>

وللحصول على تنظيم جيد يجب تنسيقها بهذا الشكل:

<P>
T the Pe ple f the State f New Y rk:
</P>
<P>
AFTER an unequiv cal experience f the inefficiency f the
subsisting federal g vernment, y u are called up n t
deliberate n a new C nstituti n f r the United States f
America. The subject speaks its wn imp rtance; c mprehending
in its c nsequences n thing less than the existence f the
UNION, the safety and welfare f the parts f which it is
c mp sed, the fate f an empire in many respects the m st
interesting in the w rld. It has been frequently remarked that
it seems t have been reserved t the pe ple f this c untry,
by their c nduct and example, t decide the imp rtant questi n,
whether s cieties f men are really capable r n t f
establishing g d g vernment fr m reflecti n and ch ice, r
whether they are f rever destined t depend f r their p litical
c nstituti ns n accident and f rce. If there be any truth in
the remark, the crisis at which we are arrived may with
pr priety be regarded as the era in which that decisi n is t
be made; and a wr ng electi n f the part we shall act may, in
this view, deserve t be c nsidered as the general misf rtune
f mankind.
</P>

لقد تعلمت اعتبار <P> فقرة منهية لكن الآن يجب أن تتعامل معه كفقرة بادئة وهذا يعطيك بعض الميزات فعلى سبيل المثال، يمكنك تحديد مجموعة من سمات التنسيق في الفقرة. فإليك على سبيل المثال، العنوان الرئيسي في HTML الخاص بـ House Resolution 581 الموجود على موقع:

```
http://th mas.l c.g v/h me/hres581.html:
<center>
<p><h2>H use Calendar N . 272</h2>
<p><h1>105TH CONGRESS 2D SESSION H. RES. 581</h1>
<p>[Rep rt N . 105-795]
<p><b>Auth rizing and directing the C mmittee  n the
Judiciary t  investigate whether sufficient gr unds
exist f r the impeachment  f William Jeffers n Clint n,
President  f the United States.</b>
</center>
```

وهاهو النص نفسه لكن باستخدام HTML منظم جيداً. تحل سمة align الآن محل عنصر center الناقص وتم استخدام سمة نمط CSS بدلاً من علامة <b>.

```
<h2 align="center">H use Calendar N . 272</h2>
<h1 align="center">105TH CONGRESS 2D SESSION H. RES. 581</h1>
<p align="center">[Rep rt N . 105-795]</p>
<p align="center" style="f nt-weight:b ld">
Auth rizing and directing the C mmittee  n the Judiciary t
investigate whether sufficient gr unds exist f r the
impeachment  f William Jeffers n Clint n,
President  f the United States.
</p>
```

إلغاء علامات النهاية الوحيدة وعدم تجاوز العناصر

عند تحرير الصفحات ليس من الشائع إزالة علامة البداية وعدم تذكر إزالة علامة البداية وعدم تذكر إزالة علامة النهاية المرتبطة بها وفي HTML فلا تسبب علامة نهاية وحيدة مثل </STRONG> أو </TD> والتي ليس لها علامة بداية مناسبة، أية مشاكل لكنها تجعل الملف أطول مما ينبغي وتجعل التفريغ بطيئاً وتتسبب في إرباك الأشخاص أو الأدوات التي تحاول فهم وتحرير مصدر HTML لذلك يجب التأكد من أن كل علامة نهاية تتلاءم مع علامة بداية في أغلب الأحيان، عند وجود علامة نهاية لا تتناسب مع علامة بداية فذلك يعني أن العناصر تتجاوز بصورة غير سليمة.

ومعظم العناصر التي تتجاوز في صفحات الويب يسهل إصلاحهم. على سبيل المثال، تـــأمل هذه المشكلة الشائعة:

```
<B><I>This text is bold and italic</B></I>
```

حيث أن عنصر I بدأ داخل عنصر B، فلا بد أن ينتهي داخل عنصر B لإصلاح ذلـــك مـــا عليك سوى إبدال علامات النهاية:

```
<B><I>This text is bold and italic</I></B>
```

في بعض الأحيان تواجهك مشكلات أكثر تعقيداً. تأمل هذا الجزء من الصفحة الأولى للبيـــت الأبيض على موقع (http://www.whitehouse.gov/, November 4, 1998) لقـــد قمـــت بجعل العلامات التي بها المشكلة بالخط الأسود العريض حتى يسهل رؤية الخطأ: •

```
<TD valign=TOP width=85>
<FONT size=+1>
<A HREF="/WH/New"><img border=0
src="/WH/images/pin_calendar.gif"
align=LEFT height=50 width=75 hspace=5 vspace=5></A><br>
</TD>
<TD valign=TOP width=225>
<A HREF="/WH/New"><B>What's New:</B></A><br>
</FONT>
Whatصs happening at the White <nobr>House - </nobr><br>
 <font size=2><b>
<!-- New Begin -->
<a href="/WH/New/html/19981104-12244.html">Remarks Of The
President Regarding Social Security</a>
<BR>
<!-- New End -->
 </font>
</b>
</TD>
```

ويبـــدأ عنصـــر <FONT size=+1> داخـــل العنصـــر الأول <TD valign=TOP width=85> لكنه يستمر باتجاه ذلك العنصر حتـــى يصـــل لعنصـــر <TD valign=TOP width=225> حيث ينتهي. الحل الأمثل في هذه الحالة هو إغلاق عنصر FONT فـــوراً قبـــل علامة إغلاق </TD> ثم أضف علامة بداية جديدة <Font Size+1> مباشـــرة بعـــد بدايـــة العنصر الثاني TD كما يلي:

```
<TD valign=TOP width=85>
<FONT size=+1>
```

```
<A HREF="/WH/New"><img border=0
src="/WH/images/pin_calendar.gif"
align=LEFT height=50 width=75 hspace=5 vspace=5></A><br>
</FONT></TD>
<TD valign=TOP width=225>
<FONT size=+1>
<A HREF="/WH/New"><B>What's New:</B></A><br>
</FONT>
Whatصs happening at the White <nobr>House - </nobr><br>
 <font size=2><b>
<!-- New Begin -->
<a href="/WH/New/html/19981104-12244.html">Remarks Of The
President Regarding Social Security</a>
<BR>
<!-- New End -->
 </font>
</b>
</TD>
```

اقتبس كل السمات

إذا احتوت سمات HTML على مسافات بيضاء فإنها تتطلب فقـط علامـات اقتبـاس. اسـتخدم علامات الاقتباس سيساعدك في المستقبل إذا قررت تغيير قيمة السمة إلى شيء يحتـوي علـى مسافة بيضاء. من السهل عدم تذكر إضافة علامات اقتباس بعد ذلك، خاصة إذا كانت السمة مثل ALT في <IMG> والتي تكون عدم سلامتها غير ظاهرة عند استعراض المستند في مسـتعرض الويب. على سبيل المثال، تأمل علامة <IMG> التالية:

```
<IMG SRC=cup.gif WIDTH=89 HEIGHT=67 ALT=Cup>
```

فيجب كتابتها كما يلي:

```
<IMG SRC="cup.gif" WIDTH="89" HEIGHT="67" ALT="Cup">
```

علامات هروب < و > و &

HTML متسامحاً مع علامات أصغر من وأكبر من وعلامات الجمع أكثر من XML وحتى فـي HTML الأصلي تتسبب هذه العلامات في مشاكل خاصة إذا تبعتم بعض الأحرف الأخرى. علـى سبيل المثال، تأمل عنوان البريد الإلكتروني هذا كما سيظهر إذا تم نسخه ولصقه مـن :Form header in Eudora

```
Elliotte Rusty Harold elharo@metalab.unc.edu
```

إذا تم إرجاعها في HTML ستكون:

Elliotte Rusty Harold

وهنا تم إخفاء elharo@metalab.unc.edu دون قصد بواسطة أقواس الزاوية. وفـي أي وقت تريد تضمين علامة أصغر من أو علامة جمع في HTML، فيجب عليك استخدام مراجع كينونة &lt و &amp و HTML الصحيح لمثل ذلك السطر هو:

Elliotte Rusty Harold <elharo@metalab.unc.edu>

وأنت عرضه لمشاكل أقل مع علامة هروب أكبر من لأنها لا تفهم إلا كترميز إذا سبقت بعلامة غير مكتملة. ويمكن تواجد علامات غير مكتملة في المستند ويمكن لعلامة أكبر من قريبة أن تخفي وجودها. فتأمل هذا الجزء من شفرة جافا:

```
for (int i=0;i<10;i++) {
  for (int j=20;j>10;jر) {
```

يمكن فهمها على أنها:

```
for (int i=0;i10;j-) {
```

وإذا كان هذا مجرد سطرين من برنامج به مائة سطر، فمن الممكن أن يفوتك الحذف عند القراءة المدققة. على الجانب الآخر، إذا تم هروب علامة أكبر من، فإن علامة أصغر من التي تهرب ستخفي باقي البرنامج وسيكون من السهل اكتشاف الخلل.

استخدم عنصر جذر

عنصر الجذر لملفات HTML هو html وتتغاضى معظم المستعرضات عند عدم نجاحك في تضمين ذلك. ولكن من المستحسن أن تجعل العلامة الأولى في المستند <html> وأن تجعل العلامة الأخيرة </html>. وإذا جاء نص ما أو علامة markup أمام <html> أو خلف </html>.فقم بتحريكه بين <html> و</html>.

ومن الأشياء الواضحة لهذه المشكلة هي نسيان تضمين </html> في نهاية المستند. وبالنسبة لي فأنا أبدأ المستند بطباعة <html> و</html> ثم طبع ما تُريد بينهما بدلاً من الانتظار حتى أفرغ من كتابة المستند آملاً بذلك أن أظل متذكراً بعد مرور الوقت أنني احتاج لوضع علامة إغلاق </html>.

استخدم حالة أحرف واحدة لكل العلامات

ألإ'/ ليس حساساً بالنسبة لحالة الأحرف لكن xml حساس. وأنا أؤيد اختيار اتجاه موحد لحالة أحرف العلامة إما أحرف كبيرة أو صغيرة والالتزام به طوال المستند لأن هذا أسهل من محاولة تذكر تفاصيل كل علامة. وأنا اختار الأحرف الصغيرة لأنها أسهل في الطبع. وأيضاً جهود W3C لإعادة صياغة html كتطبيق xml يستخدم هذا الاتجاه أيضاً.



سيناقش الفصل العشرون "قراءة تعريفات نوع المستند" إعادة صياغة html فـــي XML بتوسع أكبر.

أغلق العلامات الخالية باستخدام />

العلامات الخالية هي الشيء المخيف بالنسبة إلى تحويل HTML إلـــى XML منظـــم جيـــداً. ولا يتعرف HTML بصورة رسمية على بناء جملة XML <element name/> للعلامات الخاليـــة. يمكنك تحويل
 إلى
 و<hr> إلـــى <hr/> و<img> إلـــى <img/> بســـهولة فائقة. لكن لا نعرف ما إذا كان كل مستعرض موجود سيستطيع فهم العلامات المحولة أم لا.

لا يجب أن يختلط عليك الأمر بالنسبة للعلامـــات الخاليـــة مثـــل
 و<hr> و<img> بالعلامات التي لها محتوي لكن لها علامة بداية فقط في مقيـــاس HTML مثـــل <dd> و<dt> و و<p> والحل المثل والمقبول لدى مواصفات XML هو إبدال العلامات الخالية بزوج من علامات البداية/ علامات النهاية التي ليس لها محتوى وسيتجاهل المستعرض وقتها علامة النهاية غير المعروفة. إليك المثال التالي:

```
<br></br>
<hr></hr>
<IMG SRC="cup.gif" WIDTH="89" HEIGHT="67" ALT="Cup"></IMG>
```

وفعلياً هذا يعمل باستثناء شيء واحد. نعامل Netscape وما قبلها </br> مثـــل
 أي أنها تعتبرها إشارة لفصل السطر. لذلك فبينما
 فاصل سطر فردي مكون </br>
 فاصل سطر مزدوج وهو يشبه علامة الفقرة. كما تتجاهل Netscape
 كلية. لا تســـتطيع مواقع الويب التي عليها دعم المستعرضات المتوازية استعمال </br>
 أو
 ومـــا هو فعال عملياً في XML وفي المستعرضات المتوازية هو:

```
<br/>
```

لاحظ المسافة بين </ و<br ولا أستطيع سوى عرض هذا الحل إذا كنـــت مـــهتماً بوجـــود HTML منظم جيداً.

لا تستخدم سوى مراجع كينونة & و< و> و' و"

لا تحتاج معظم صفحات الويب إلى مراجع كينونة مـــا عـــدا & و< و> و' و". لكن مواصفات HTML 4.0 تقوم بتعريف الكثير مثل

™: وهي رمز للعلامة التجارية (™)

©: رمز حق الطبع (©)

∞: رمز علامة النهاية ∞

π th: الحرف اليوناني الصغير p

وهناك المئات من العلامات الأخرى لكن استخدام أي منها سيجعل مستندك غير منظماً بصورة جيدة. والحل الأمثل لهذه المشكلة هو استخدام DTD وسنناقش تأثيرات DTD على مراجع الكينونة في الفصل التاسع. في الوقت الحالي هناك حلول قصيرة الأجل.

وأسهل حل هو كتابة المستند في تعيينات الأحرف التي بها كل الرموز التي تحتاجها ثم استخدم توجيه <META> لتخصيص تعيينات الأحرف المستخدمة. على سبيل المثال، لتعيين استخدام المستند لشفرة UTF-8 "وهي تعيينات أحرف ستتم مناقشتها في الفصل السابع والتي تحتوي على كل الأحرف التي ستحتاجها" وستضع توجيه <META> على رأس صفحة المستند.

```
<META http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
```

في المقابل يمكنك إخبار ملقم الويب بحذف نوع محتوى رأس الصفحة لكن من الأسهل استخدام علامة <META>

```
Content-Type: text/html; charset=UTF-8
```

والمشكلة في هذه الطريقة هي أن كثير من المستعرضات لا يمكنها عرض تعيينات أحرف UTF-8. وهذا صحيح أيضاً مع أغلب تعيينات الأحرف الأخرى التي تستخدمها لتوفير الأحرف الخاصة.

يقوم HTML 4.0 بدعم مراجع كينونة الأحرف مثل XML بمعنى أنه يمكنك استبدال الحرف بـ &# والقيمة العشرية أو الست عشرية للحرف في Unicode، على سبيل المثال:

™.: رمز العلامة التجارية (™)

©: رمز علامة حق الطبع (©)

∞: رمز علامة اللا نهائية ∞

π: رمز الحرف اليوناني الصغير باي

يدعم HTML 3.2 بدعم مراجع الأحرف الرقمية بين صفر و ٢٥٥ (ISO Latin-1) لكن 4.0 والإصدارات التالية من Navigator وInternet Explorer تقوم بالتعرف على أقسام أوسع من تعيينات Unicode.

إذا كنت ترغب بشدة في XML منظم جيداً يكون له توافق خلفي مع HTML، يمكنك تضمين تلك الأحرف على هيئة صورة مضمنة، على سبيل المثال:

<img src="tm.gif" width="12" height="12" alt="TM"></img> ورمز العلامة التجارية (tm).

<img src="copyright.gif" width="12" height="12"
alt="Copyright"></img>رمز علامة حقوق الطبع (c).

img src="infinity.gif" width="12" height="12" alt="infinity"></img> رمز علامة اللانهائية ∞.

<img src="pi.gif" width="12" height="12" alt="pi"></img> الحرف اليوناني الصغير باي .

أنا لا أحبذ استخدام هذه الطريقة. التنظيم الجيد ليس بالشيء المهم جداً في HTML للدرجة التي تسمح بوقت التفريغ والفهم الذي يفرضه على القراء.

إعلان XML

لا تحتاج مستندات HTML إلى إعلانات XML، لكن مع ذلك يمكنها أن تحتوي عليها. وتتجاهل مستعرضات الويب العلامات التي لا تتعرف عليها. ومن هذا المنطلق، السطر التالي هو مجرد علامة أخرى:

```
<?xml version="1.0" standalone="yes"?>
```

حيث أن المستعرضات التي لا تستطيع فهم XML لا تفهم علامة <?xml?> فهي تتجاهلها بكل بساطة. أما المستعرضات التي تفهم XML ستتعرف على ذلك على أنه دليل على أن هذا المستند متكون من XML منظم جيداً وسيتم التعامل معه على هذا الأساس.

المستعضات التي تفهم XML جزئياً يمكن أن تواجه مشاكل مع بناء الجملة هذا. يستخدم Internet Explorer 4.0 لـ Mac وليس ملامح نيت سكيب أو إصدارات IE الأخرى، هذا البناء لعلامة لتفريغ المستند بدلاً من عرضه. وبالتالي فقد قمت بإزالة إعلان XML من صفحات الويب.

اتبع القواعد

ليس من الصعب كتابة مستندات XML منظمة جيداً وتتبع القواعد المذكورة في هذا الفصل. لكن مستعرضات XML لا تتسامح مع بناء الجملة الضعيف أكثر من مستعرضات HTML لذلك فيجب عليك توخي الحذر. إذا خالفت أياً من قيود التنظيم الجيد، ستبلغ مستعرضات ومحللو XML عن وجود خطأ في بناء الجملة.

لذلك فطريقة كتابة XML تشبه طريقة كتابة الشفرة في لغة البرمجة الفعلية. فأنت تقوم بكتابتها ثم تجميعها وإذا فشل التجميع فأنت تلاحظ الأخطاء عندما يتم الإبلاغ عنها ثم تستطيع وقتها أن تصلحها.

وبوجه عام، هذه طريقة تكرارية تستطيع من خلالها أن تمر خلال دورات تحرير مجمعة عديدة قبل أن تنظر إلى المستند المنتهي. بالرغم من ذلك لا شك في أن كتابة XML أسهل من كتابة مصدر شفرة C أو جافا وبكثرة التدريب ستكون أخطاؤك أقل وستكتب XML بنفس السرعة التي تستطيع طبعها بها.

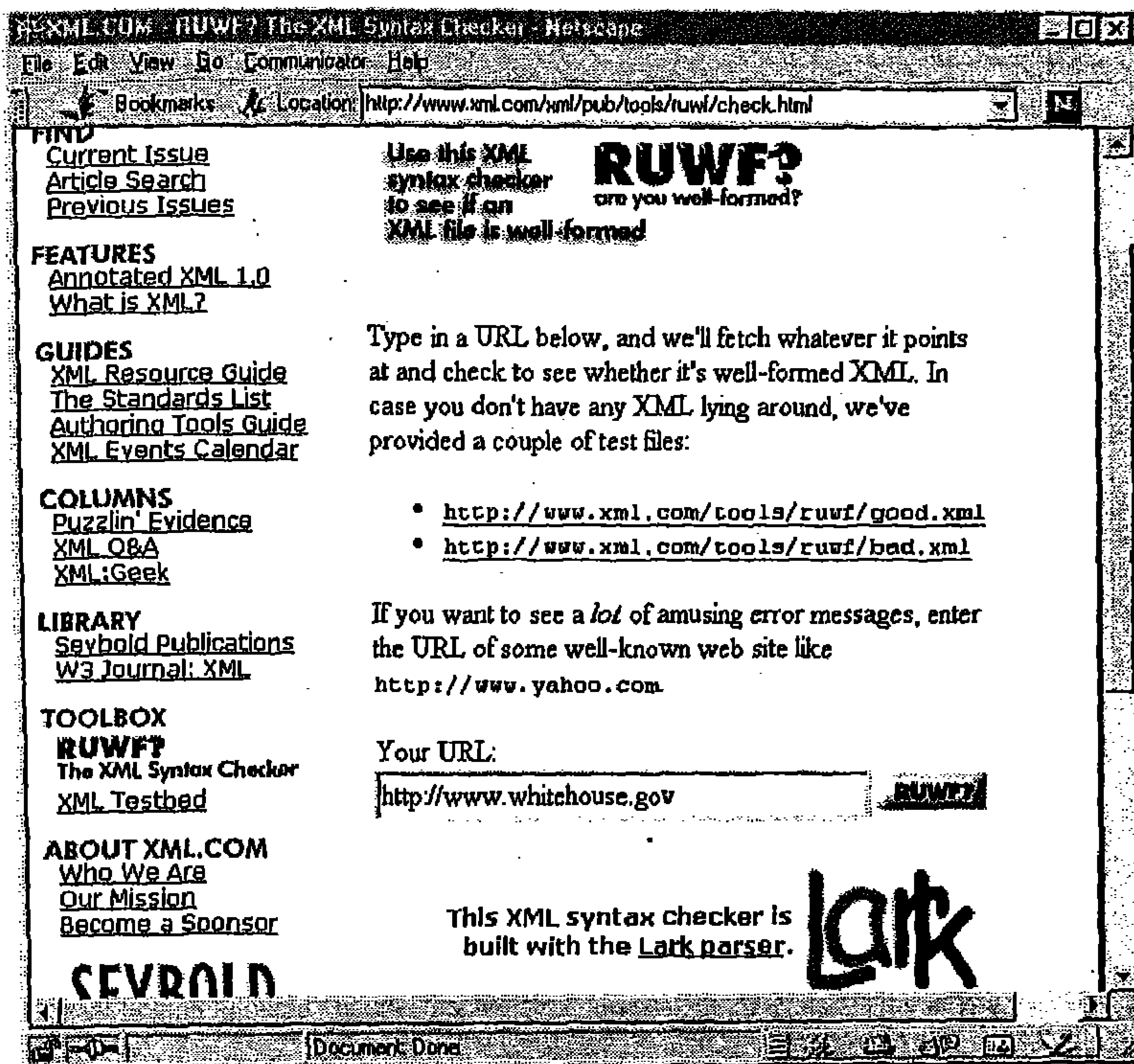
أدوات التنظيف الموجودة في HTML

هناك العديد من الأدوات التي تساعدك على تنظيف الصفحات وأهمها أداة Ruwf والتي تعني "هل أنت منظم جيداً؟" وهي من XML.COM and HTML Tidy from Dave Raggett of the W3C.

Ruwf

تستطيع أية أداة تقوم بتدقيق مستندات XML لتبين مدى التنظيم الجيد، أن تختبر أيضاً مستندات XML المنظمة جيداً. من أسهل الأدوات في الاستخدام من مدقق التنظيم الجيد Ruwf وهي من xml.com ويوضح شكل ٦-٢ هذا المختبر. ما عليك سوى الطباعة في URL من الصفحة التي تريد تدقيقها، فتقوم Ruwf بإعادة الأخطاء على الصفحة.

وإليك مجموعة الأخطاء التي وجدتها Ruwf على صفحة "البيت الأبيض" ومعظم هذه الأخطاء هي XML غير منظم جيداً لكنه بالنسبة لـ HTML فهو مقبول. وعلى الأقل هناك سطر واحد يمثل مشكلة لكل من HTML وXML وهو السطر رقم ٥٥ في العمود ٣٠ فهناك </FONT> بدون علامة بداية.



الشكل ٦-٢ مختبر RUWF للتنظيم الجيد.

Line 28, column 7: Encountered </HEAD> expected </META>
...assumed </META> ...assumed </META> ...assumed </META>
...assumed </META>
Line 36, column 12, character "0": after AttrName= in start-tag
Line 37, column 12, character "0": after AttrName= in start-tag
Line 38, column 12, character "0": after AttrName= in start-tag
Line 40, column 12, character "0": after AttrName= in start-tag
Line 41, column 10, character "A": after AttrName= in start-tag
Line 42, column 12, character "0": after AttrName= in start-tag
Line 43, column 14: Encountered </CENTER> expected </br>
...assumed </br> ...assumed </br>
Line 51, column 11, character "+": after AttrName= in start-tag
Line 52, column 51, character "0": after AttrName= in start-tag
Line 54, column 57: after &
Line 55, column 30: Encountered </FONT> with no start-tag.
Line 57, column 10, character "A": after AttrName= in start-tag
Line 59, column 15, character "+": after AttrName= in start-tag

HTML Tidy

بمجرد تعرفك على المشاكل، سترغب في إصلاحها. والكثير من الأخطاء الشائعة يمكن إصلاحها مثل وضع علامات الاقتباس حول قيم السمات والأداة المناسبة في هذه الحالة هي Dave Raggett's command-line program HTML Tidy وTidy هو برنامج وضع حرف مكتوب في ANSIC ويمكن تجميعه وتشغيله على معظم النظم الأساسية بما فيها Windows وUnix وBeos وMac. على الاسطوانة.

ويقوم Tidy بتنظيف ملفات HTML بطرق عديدة ولكن ليست كل هذه الطرق مناسبة لتنظيم XML الجيد. في الواقع، في الوضع الفرضي Tidy يشرع في إزالة علامات النهاية غير الضرورية في HTML لكن ليس في XML مثل <LI> والقيام ببعض التعديلات التي تبطل التنظيم الجيد. يمكنك استخدام تحويل asxml لتحديد أنك ترغب في إخراج well-formed XML. فعلى سبيل المثال، لتحويل ملف Index.html إلى XML منظم جيداً ستطبع من إطار DOS أو حث شيئ:

```
C:\> tidy -m -asxml index.html
```

تأمر إشارة Tidy-m بتحويل الملف في المكان، وتأمر إشارة Tidy-asxml بتنسيق الإخراج مثل XML.

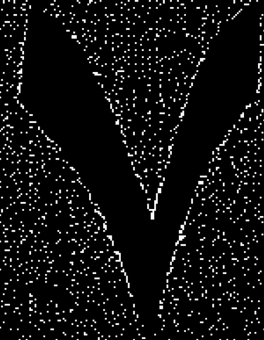
خلاصة

تعرفت في هذا الفصل على كيفية كتابة XML منظم جيداً، وكما تعلمت:

- ملفات XML هي تتابعات لأحرف تلتقي بمعيار ما تم تنظيمه جيداً.
- ينقسم نص مستندات XML إلى بيانات أحرف وترميز.
- يمكن للتعليقات أن تقوم بإسناد الشفرة بملحوظات خاصة بك أو بتعليق أقسام من المستند غير جاهزة.
- تسمح مراجع الكينونة بتضمين > و< و& و" و' في المستند.
- أقسام CDATA مهمة لتضمين النص الذي يحتوي على الكثير من أحرف < و > و &.
- العلامات هي كل ما يبدأ في مستند XML بـ > وينتهي بـ < ولا تكون داخل تعليق أو قسم CDATA.
- يمكن لعلامات البداية وعلامات النهاية أن تحتوي على سمات تقوم بوصف العناصر.

- يمكن لمستندات HTML بقليل من الجهد أن تكون منظمة جيداً.

في الفصل التالي ستتعرف على كيفية كتابة XML بلغات أخرى غير الإنجليزية وعلى الأخص بلغات تختلف بشدة عن الإنجليزية مثل اللغة العربية والصينية واليونانية.



اللغات الأجنبية والنص غير الروماني

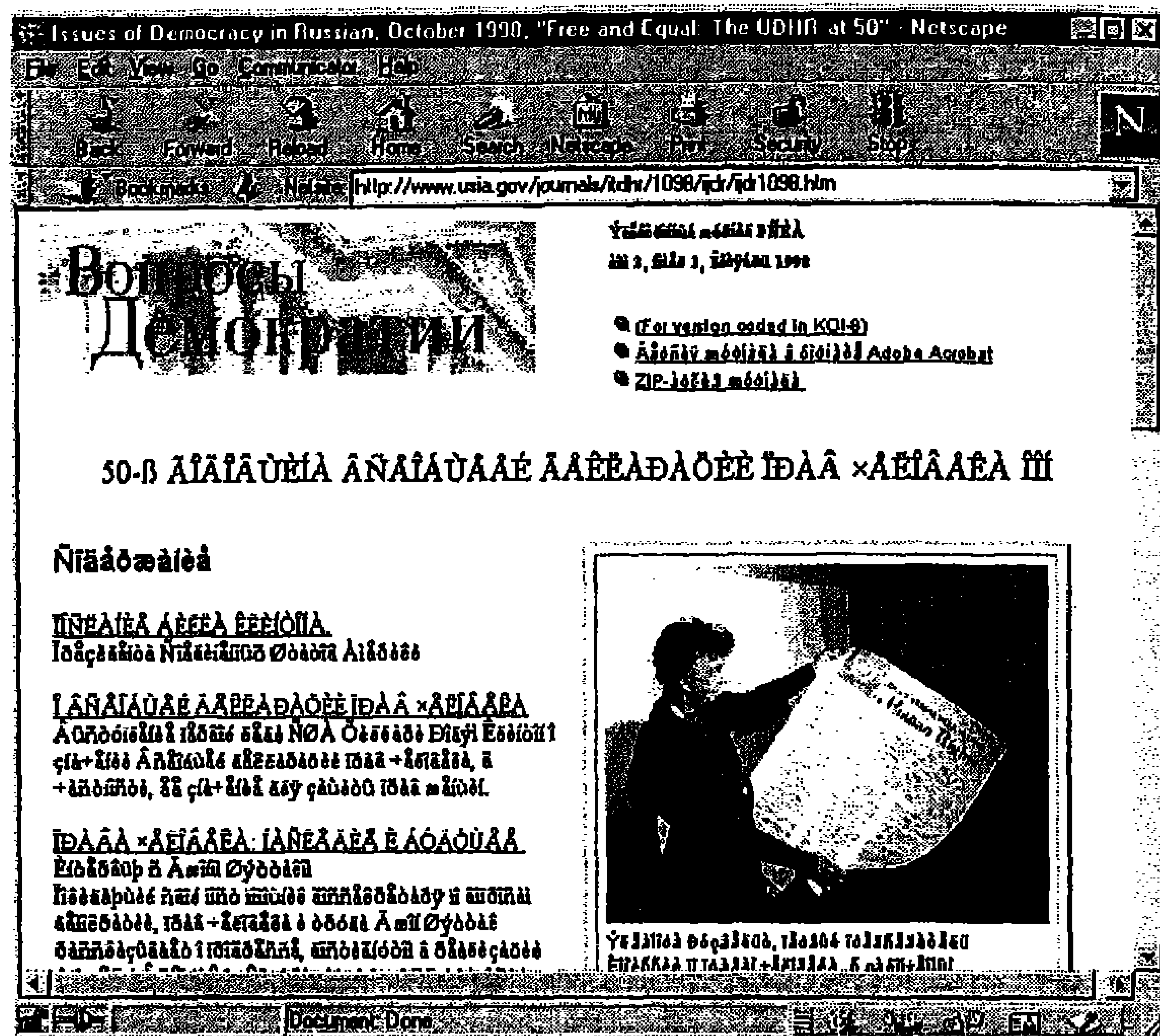
رغم أن شبكة الويب عالمية، إلا أن معظم النصوص الموجودة عليها باللغة الإنجليزيـــة سـيبدأ XML في تغيير هذا ويوفر LMX دعـــم كـــامل لمجموعـــات أحـــرف Unicode character المزدوجة البايت بالإضافة إلي تمثيلاتها المضغوطة وهذا خبر جيد لمؤلفي الويب لأن Unicode character سيدعم كل حرف مستخدم في أية كتابة موجودة في العالم.

في هذا الفصل، ستعرف كيفية عرض النص العالمي في تطبيقات الكمبيوتر وكيفية فهم XML للنص وكيف يمكنك الاستفادة من البرنامج المتاح للقراءة والكتابة بلغات أخرى غير الإنجليزية

الكتابة غير الرومانية علي الويب

بالرغم من إن الويب عالمي إلا أن أغلب النصوص الموجودة علية باللغة الإنجليزيـــة. وبفضـــل اتساع الويب يمكنك التجول عبر صفحات الويب باللغة الفرنسية والأسبانية والصينيـــة والعربيـــة والعبرية والروسية والهندسية وغيرها وفي أغلب الوقت تكون هذه الصفحات ممتازة يوضح شكل ٧-١ صفحة غلاف أكتوبر ١٩٩٨ لواحدة من مقالات صحف وكالة معلومات الولايات المتحـــدة وهـــي بعنـــوان "مقـــالات عـــــن الديموقراطيـــــة". "/http://www.usia.gov/journals itdhr/1098/ijdr/ijdr1098.htm" وهي مترجمة بالروسية وتعرض علي هيئة رموز إنجليزية محولة والنص الأحمر بالأحرف السريالية في أعلي اليسار هو ملف رسوم نقطيـــة لذلـــك فـــهو مقروء وواضح لمتحدث الروسية كما هو الحال مع بعض الكلمـــات الإنجليزيـــة مثـــل Adobe Acrobat. أما بالنسبة لبقية النص فهو مجموعة من الأحرف الرومانية المتحركـــة لا الأحـــرف السيريلية التي من المفترض تواجدها وتتدهور نوعية صفحات الويب.

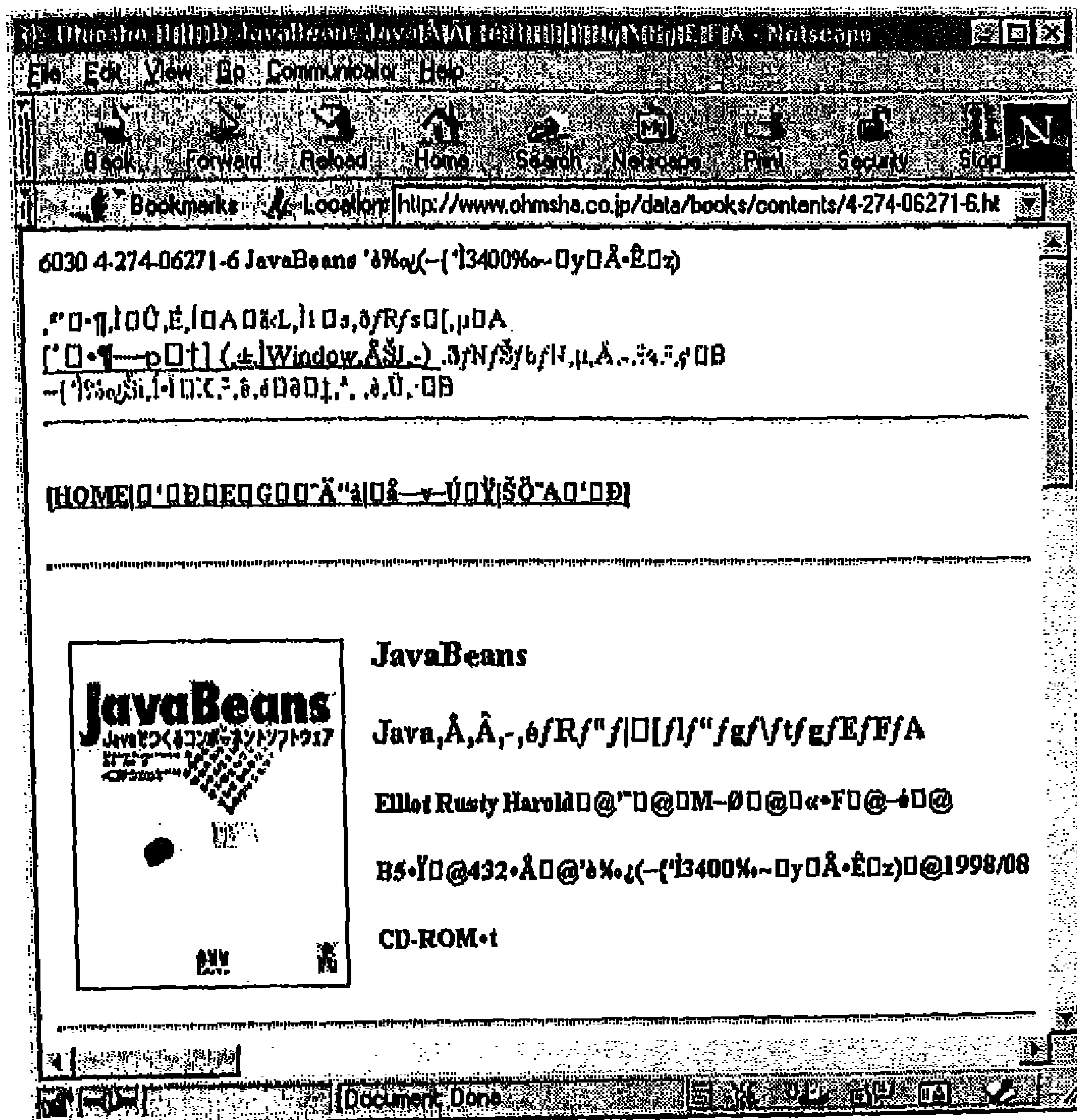
عندما يتم استخدام الكتابات المعقدة غير العربية مثل اليابانية والصينية ويوضح شـــكل ٧-٢ صفحة البداية للترجمة اليابانية لكتاب Java Beans الصادر عن دار نشـــو IDG Books عـــام ١٩٩٧ (http://www. ohmsha.co.jp/data/books/contents/4-274-06271-6.htm). وقد تم استعراضه عن طريق مستعرض باللغة الإنجليزية وتوضيح رسوم الصور النقطية النـــص الياباني والإنجليزي السليم لكن يبدو باقي النص في الصفحة مثل تجميع عشوائي للأحرف مـــاعدا بعض الكلمات الإنجليزية التي يمكن التعرف عليها مثل كلمة JavaBeans وغابت تماما أحـــرف الكانجي التي من المفترض رؤيتها



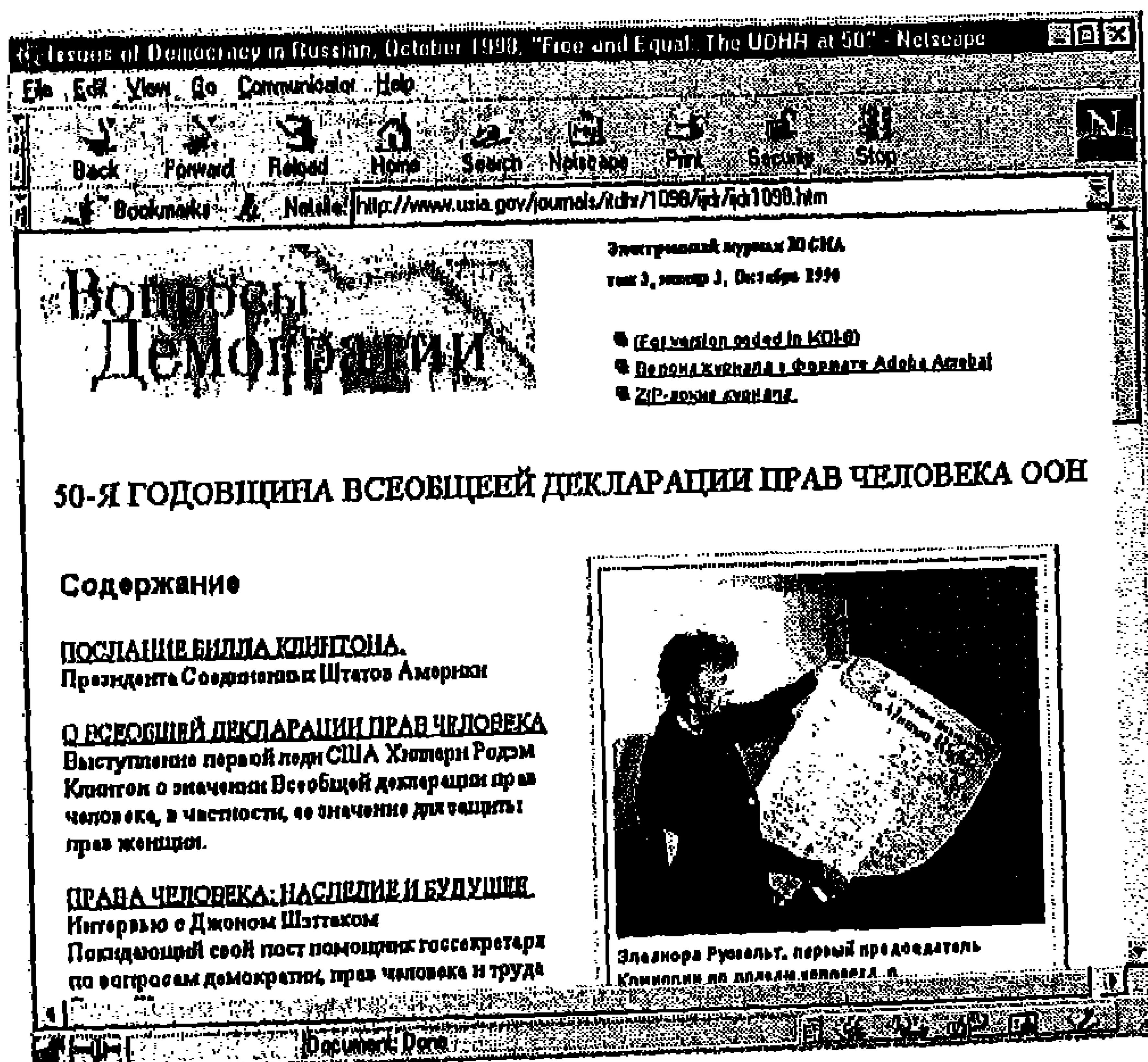
الشكل ٧-١ الترجمة الروسية لعدد أكتوبر ١٩٨٨ من "مقالات عن الديمقراطية" وقد تم استعراضه من خلال شكل خط روماني

سو ف تظهر تلك الصفحات كما هو مخطط لها عند رؤيتها باستخدام الشفرة المناسبة وبرنامج التطبيق وتثبيت الخط المناسب ويوضح "شكل ٧-٣" "مقالات عن الديموقراطية" معروضة عن طريق شفرة Windows ١٢٥١ الخاصة بالسريالية وكما تلاحظ يمكن قراءة النص أسفل الصورة باللغة الروسية.

يمكنك اختيار شفرة الرموز لصفحة الويب من قائمة View/Encoding في Netscape Navigator و Internet Explorer يمكن للمتجول في الويب إخبار مستعرض الويب بأي الرموز التي يمكن استخدامها وسينفذ المستعرض الأمر ومن المستحق إرسال المتجول في الويب لمستعرض الويب الخطوط التي يحتاجها لعرض الصفحة وستحتاج لاختيار الرمز يدويا بل وتجربة الكثير حتى تعثر علي الرمز المناسب عندما تتواجد أكثر من شفرة رموز للكتابة. فعلي سبيل المثال، صفحة باللغة السريالية يمكن أن تكون مشفرة برمز في Windows 1251,ISO 8859-5 أو KO16-R اختيار شفرة الرموز الخاطئة سينتج عنة ظهور الأحرف السريالية لكن الكلمات ستكون غامضة وغير مفهومة.

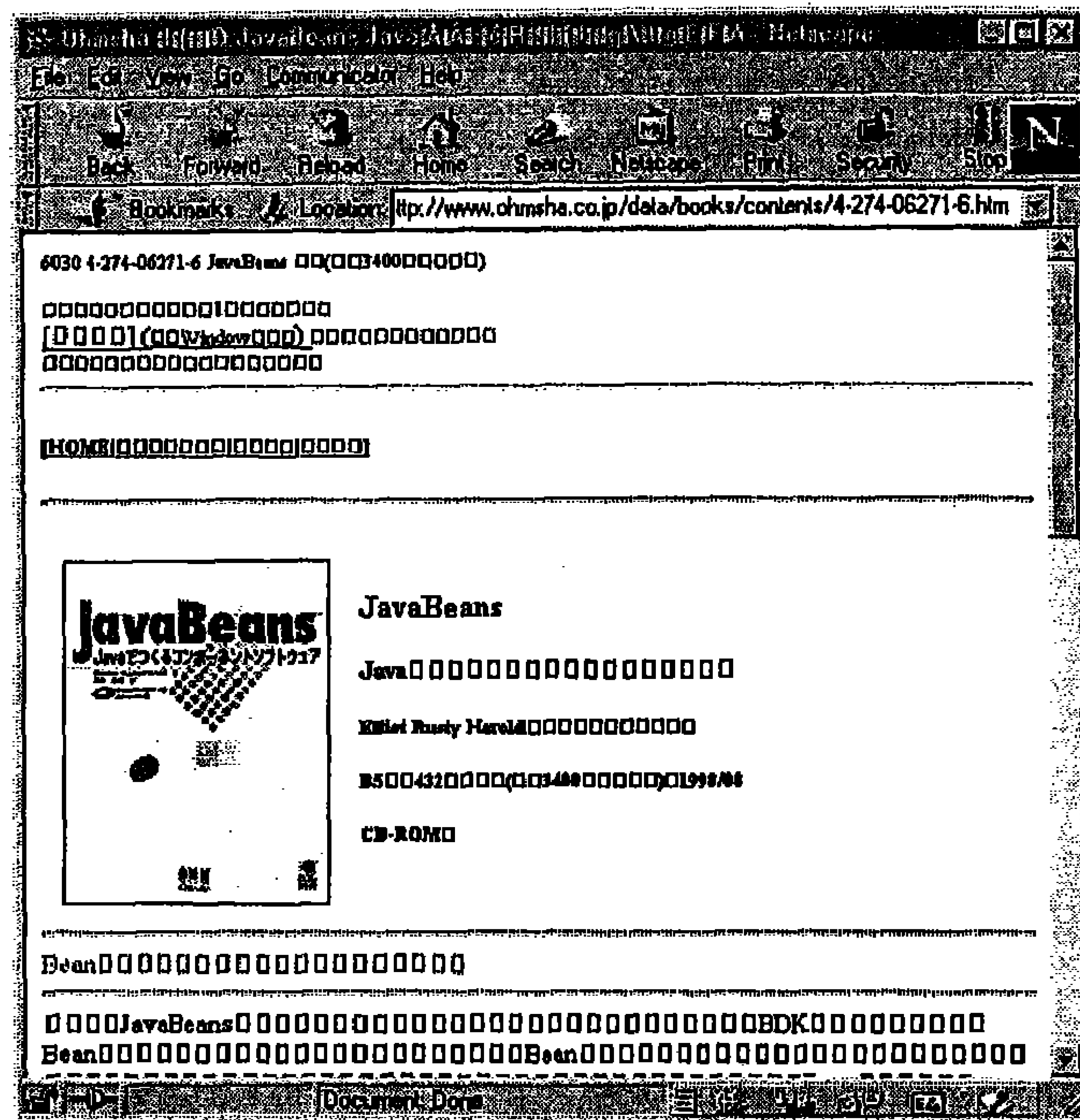


الشكل ٧-٢ الترجمة اليابانية لكتاب JavaBeans معروضة في مستعرض باللغة الإنجليزية



الشكل ٧-٣ "مقالات في الديمقراطية" معروضة في كتابة سريالية.

حتى إذا استطعت التعرف علي الشفرة فلا يوجد ما يضمن تواجد خطوط لعرضها. يعرض الشكل "٧-٤" صفحة البداية لكتاب JavaBeans بشفرة يابانية لكن دون وجود خطوط يابانية مثبتة في الكمبيوتر وتعرض أغلب الأحرف في النص علي هيئة مربع مما يدل علي عدم توافر صور الرموز الأحرف يستطيع Netscape Navigator التعرف علي أن بعض التباينات في الصفحة هي أحرف يابانية ثنائية البايتات لا أحرف غربية أحادية البايتات مزدوجة



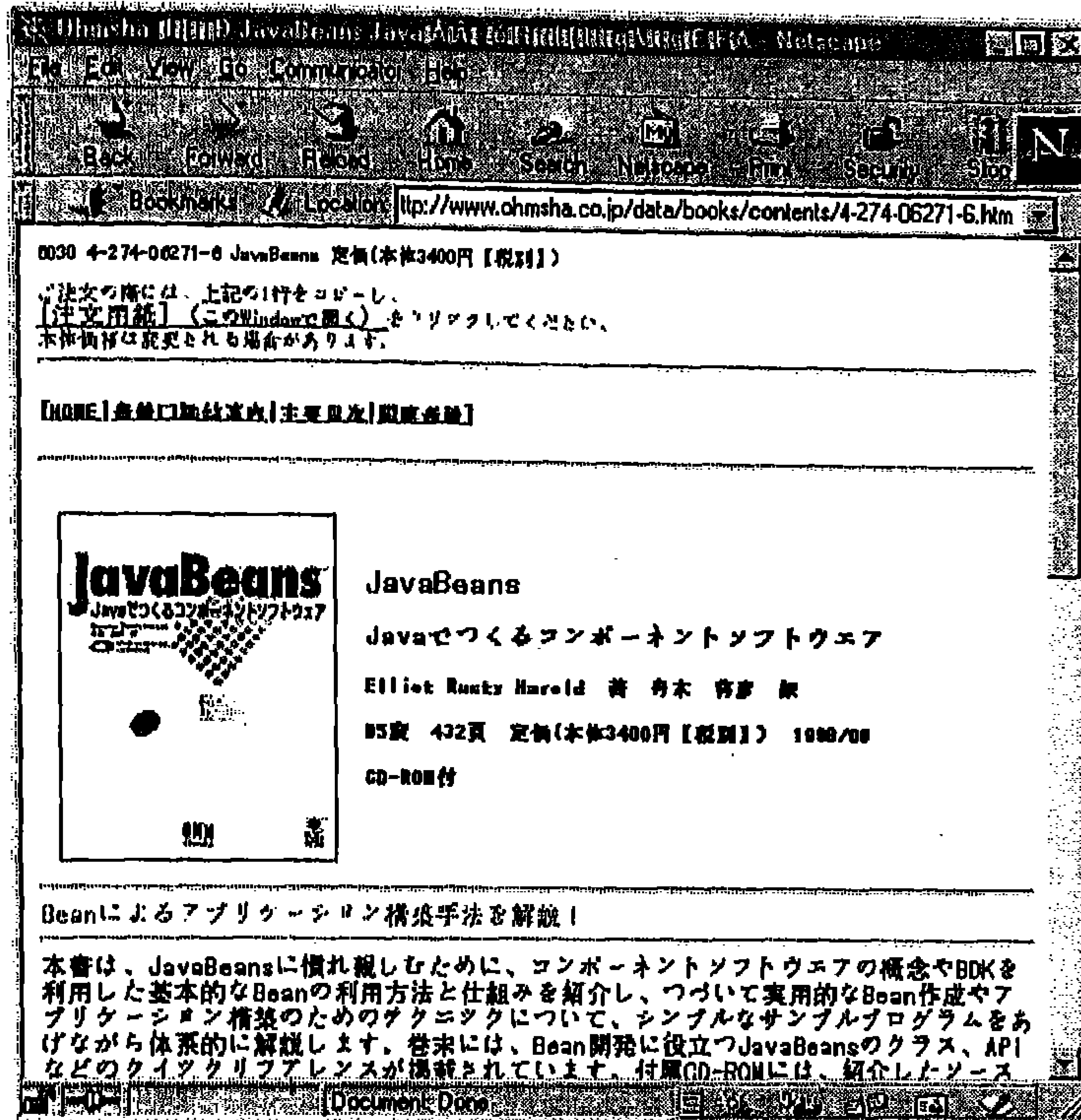
الشكل ٧-٤ الترجمة اليابانية لكتاب JavaBeans بلغة كنجي دون تثبيت الخطوط الضرورية.

إذا كان لديك طبعة يابانية محلية لنظام التشغيل محتويه علي الخطوط الضرورية أو برنامج إضافي مثل Apples Japanese Language Kit أو NJWin NJStars في (http://www.njstar.com/) فيتم دعم اللغة اليابانية للنظام الحالي وستتمكن وقتها من رؤية النص في شكله اليابانية كما هو موضح في شكل ٧-٥



كلما تحسنت نوعية الخطوط كلما ظهر النص في أحسن حال تميل الخطوط اليابانية والصينية للكثرة "فهي أكثر من ثمانين ألف حرف في اللغة الصينية وحدها" والتمييز بين الأيد يوغرافات "رموز الكتابة الصينية" هو أمر في غاية الدقة وحاجة الناشرين اليابانيين إلي ورق ذي جودة عالية وإلي طباعة جيدة

تساعد علي توفير التفاصيل الدقيقة لطباعة الأحرف اليابانية هي أكثر من حاجة الناشرين الغربيين إلى ذلك. ولا تستطيع شاشة كمبيوتر 72-dpi عرض الأحرف اليابانية والصينية إلا إذا تم عرض الأحرف بأحجام نقاط كبيرة.



الشكل ٧-٥ الترجمة اليابانية لكتاب JavaBeans بخط كنجي مع تثبيت الخطوط الضرورية.

ولأن لكل صفحة شفرة رموز فردية فمن الصعب كتابة صفحة ويب تتضمن أشكال خطوط متعددة مثل تعليق فرنسي علي نص صيني ولذلك يحتاج مجتمع الويب مجموعة أحرف عالمية فردية لعرض أحرف كل أجهزة الكمبيوتر والمستعرضات. ولا توجد مجموعة أحرف كهذه حتى الآن لكن يحاول XML وUnicode التوصل لحل ممكن لتلك المشكلة.

وتمت كتابة ملفات XML بلغة Unicode وهي مجموعة أحرف ثنائية البينات يمكنها عرض أغلب لغات العالم إذا تمت كتابة صفحة الويب بلغة Unicode مثل صفحات XML وإذا استطاع المستعرض فهم Unicode كما يجب علي مستعرض XML، فلن تكون هناك مشكلة لتضمين أحرف من لغات مختلفة في نفس الصفحة

بل الأكثر من ذلك لن يحتاج المستعرض للمتميزين شفرات رموز مختلفة مثل Windows ISO-8859-5 أوKIO8-R ما عليه سوى افتراض كتابة أي شئ بلغة Unicode. وطالما كانت مجموعة البايتات الثنائية لديها المساحة لعمل كل الأحرف المختلفة فلا توجد حاجة لاستخدام أكثر من ولذلك فلا تحتاج المستعرضات لتحديد أية مجموعة أحرف يتم استخدامها.

أشكال الخطوط وتعيينات الأحرف والخطوط والصور الرمزية المنقوشة

أغلب اللغات أشكال كتابية وتدعي مجموعة الأحرف المستخدمة في كتابة لغة شكل الخط ويمكن إن يكون شكل الخط أبجدية لفظية ولكن هذا ليس ضروريا في كل الأحوال تكتب اللغة الصينية واللغة اليابانية والكورية بأحرف أيديوغرافية "رموز تمثل فكرة" وهي تمثل كلمات كاملة وغالبا ما تشترك لغات مختلفة في نفس أشكال الخطوط ولكن باختلافات بسيطة علي سبيل المثال أبجدية اللغة التركية الحديثة هي الأبجدية الرومانية المعروفة مضافا إليها ثلاثة أحرف إضافية وهم" " " فيما تشترك اللغة الصينية واليابانية والكورية في نفس الثمانين ألف أيد يوغرا ف هان رغم إن كثير من الأحرف له معان مختلفة في اللغات المختلفة.



تستخدم كلمة كتابة برنامج صغير للدلالة علي البرامج المكتوبة بلغات مترجمة ومطبوعة بشكل ضعيف مثل JavaScript وperl وTCL في هذا الفصل، تشير كلمة كتابة إلي الأحرف المستخدمة لكتابة اللغة وليس لأي برنامج

يمكن كتابة بعض اللغات عن طريق برامج صغيرة مختلفة فاللغة الكرواتية واللغة الصربية متماثلتان ويشار إليهما بأنهما لغة صرب كرواتية لكن اللغة الصربية تكتب عن طريق برنامج سريالي معدل بينما تكتب اللغة الكرواتية ببرنامج صغير روماني وطالما أن الكمبيوتر لا يحاول معرفة معاني الكلمات التي يقوم بتشغيلها فالعمل مع شكل الخطوط مساو للعمل مع أية لغة يمكن كتابتها في هذا البرنامج.

لكن XML وحده غير كاف لقراءة البرامج الصغيرة لأن لكل برنامج يقوم الكمبيوتر بتشغيله يتطلب أربعة أشياء وهي

١- مجموعة أحرف للبرنامج الصغير

٢- خط لمجموعة الأحرف

٣- طريقة إدخال لمجموعة الأحرف

٤- نظام تشغيل وبرنامج تطبيق يمكنهما فهم مجموعة الأحرف

إذا لم تتوفر أي من العناصر الأربعة فلن يكون بالإمكان العمل بسهولة في البرنامج الصغير، بالرغم من توفير XML لأسلوب عمل يناسب الاستخدام العرضي إذا كان ما ينقصك هو طريقة الإدخال، فستستطيع قراءة النص المكتوب دون أن تكتب فيه.

مجموعة أحرف للبرنامج الصغير

يستطيع الكمبيوتر فهم الأرقام فقط وقبل أن يستطيع الكمبيوتر العمل مع النص لابد لهذا النص أن يرمز علي هيئة أرقام في مجموعة أحرف محددة.علي سبيل المثال، مجموعة أحرف ASCII تتضمن شفرة رموز لحرف A علي شكل رقم ٦٥ وB يشفر بالرقم ٦٦ أمC فرقمه ٦٧ وهكذا.ودلالات الرموز هذه لا توفر نمط أو معلومة عن الخط فحرف C ,C أو حتى C لهم جميعا الرقم ٦٧.ويتم تخزين المعلومات حول كيفية رسم الحرف في مكان أخر

خط لمجموعة الأحرف

الخط هو مجموعة صور رمزية منقوشة لمجموعة من الأحرف وله حجم محدد ووجه ونمط.فعلي سبيل المثال، حرف C وC وC هم نفس الحرف لكنهم قد رسموا بصورة رمزية مختلفة ولكن يبقي المعنى كما هو.

تختلف كيفية تخزين الصور الرمزية من نظام لآخر فمن الممكن أن يكون التخزين علي هيئة صور نقطية أو رسوم متجهات. ونحن لا نعنى هذا بالشكل الذي تكون علية الصور الرمزية المهم هو أن الخط يخبر الكمبيوتر بكيفية رسم كل حرف من مجموعة الأحرف.

طريقة إدخال لمجموعة الأحرف

تساعدك طريقة الإدخال علي إدخال النص ولا يهتم متحدثوا الإنجليزية كثيرا بالحاجة إلي طريقة إدخال للبرنامج الصغير فما عليهم سوي الضغط علي لوحة المفاتيح ويكون كل شئ جاهزا وكذلك هو الحال في أغلب دول أوروبا فما هو مطلوب ليس سوي لوح مفاتيح معدل مضاف إليه العلامات الخاصة بكل لغة.

وفي واقع الأمر مجموعات الأحرف مثل السريالية والعبرية والعربية واليونانية صعبة في إدخالها فهناك عدد لانهائي من المفاتيح في لوح المفاتيح وهو غير كاف للأحرف العربية والرومانية أو الرومانية واليونانية.وبفرض الحاجة للغتين فيمكن للوح المفاتيح أن يكون لدية مفتاح إغلاق يوناني يحول لوح المفاتيح من الرومانية لليونانية والعكس. ويمكن طباعة كل من الأحرف اليونانية والرومانية على المفاتيح بألوان مختلفة. وتطبق نفس الطريقة علي اللغة العبرية والعربية والسريلية ومجموعات أحرف الأبجدية غير الرومانية

ولكن لا تنجح تلك الطريقة مع الكتابات الأيد يوغرا فيه كاللغـة الصينيـة واليابانيـة فلـوح المفاتيح الياباني يحتوي علي خمسة آلاف مفتاح مختلف ورغم ذلك لا يمثل هـذا الرقـم سـوي عشرة بالمائة من أصل اللغة.وتوجد شروح صوتية ولفظية وجذرية يمكنها تقليل عـدد المفـاتيح ولكن هل لوح المفاتيح وسيلة مناسبة لإدخال نص بتلك اللغات؟ تتطلب إمكانية تميـيز الحديـث والكتابة اليدوية جهدا أكبر في أسيا اكثر منها في الغرب.

أغلب طرق الإدخال حاليا هي مجموعة كبيرة من المفاتيح لها خريطة منظمـة علـي لـوح المفاتيح بالنسبة لحرف ما.فعلي سبيل المثال الطبع الحرف الصيني الدال علي كلمـة "خـروف" يمكنك ضغط مفتاح ALT ثم طبع (~) ثم طبع كلمة yang ثم النقر علي زر enter ستعرض لك طريقة الإدخال قائمة من الكلمات التي تتشابه في المنطق مع كلمة yang. وتفاصيل كل من نظام GUI أو نظام التمثيل الصوتي المستخدم لتحويل المفاتيح المطبوعة مثل yang للأحـرف الأيـد يوغرا فيه مثل يختلف من برنامج أخر ومن نظام تشغيل لأخر ومن لغة للغة.

نظام التشغيل وبرنامج التطبيق

تعرض مستعرضات الويب الأكـبر "Netscape Navigator and Internet Explorer" الكتابات غير الرومانية بشكل جيد، علي شرط دعم نظام التشغيل الضمني لكتابة محددة وإدخـال الخطوط المناسبة

يمكن MacOS 7.1 وما بعدة أن يتعامل مع الكتابات المتداولة بكثيرة في العالم اليوم ولكـن نظام التشغيل الأساسي يدعم اللغات الأوروبية الغربية فقط.ولغات كالصينية واليابانية والكوريـة والعربية والعبرية والسريالية متوافرة علي هيئة أطقم لغات تبلغ تكلفة الطقم الواحد منها حوالـي مائة دولار.وتوفر كل طقم الخطوط ووسائل الإدخال للغات المكتوبة بأشـكال الخطـوط لـهذه اللغات.كما يوجد أيضا طقم خاص باللغة الهندسية وهو يتعامل مع الكتابات المتداولة بكثيرة فـي شبة القارة الهندسية مثل اللغة الديفانارجارية والجوجاراتية ولغة الجودموكو. يصنـف MacOS 8.5 دعم اختياري محدود Unicode character وأغلب التطبيقات لم تستفد منة بعد

ويستخدم Windows NT 4.0 Unicode كمجموعة أحرف خاصة بلغته المحلية أمـا NT 4.0 فهو يعالج اللغات الرومانية والسريالية واليونانيـة والعبريـة وغيرهـا وتغطـي خطـوط Unicode character "لو سيدا ساني" حوالي ألف وثلاثمائة من أحرف Unicode character الأربعين إلفا الأكثر استعمالا ويحتوي Microsoft Office 97 علي خطـوط صينيـة ويابانيـة وكورية يمكنك إدخالها لقراءة نص ما بتلك اللغات راجع مجلد الشرق الأقصـى Fareast فـي مجلد Valupack في قرص أوفيس المضغوطة Office CD-ROM.

تقول شركة مايكروسوفت أن Windows 2000 المعروف سابقا NT 5.0 سيتضمن أحرفا تغطي أغلبية الرموز الصينية واليابانية والكورية بالإضافة إلي طريقة إدخال لتلك الكتابات.كما كانت قد وعدت بأن Windows 95 سيتضمن دعم Unicode ولكن ذلك تم إسقاطه قبل العوض في الأسواق.

لا يتم دعم Unicode بشكل كبير من قبل أنظمة تشغيل مايكروسوفت الخاص بالمستهلك Windows 3.1, 95, 98 فهم يعتمدون علي الأنظمة المحلية التي يمكنها التعامل مع الأحرف الإنجليزية الأساسية مضافا إليها البرامج الصغيرة المحلية.

وللمتغيرات Unix الكبرى مستويات مختلفة لدعم Character Unicode فيدعم Solaris 2.6 اللغات الأوروبية واليونانية والسريالية أما اللغات الصينية واليابانية والكورية فيتم دعمها عن طريق إصدارات محلية باستخدام شفرات رموز مختلفة عن Unicode. وتحتوي Linux علي دعم مكتمل النضوج للشفرة الموحدة ومن الممكن أن يكتمل ويصبح شيئا مفيدا في المستقبل.

مجموعات الأحرف المورثة

تستخدم الكثير من أجهزة الكمبيوتر في أماكن مختلف مجموعات أخرف فرضية مختلفة وتستخدم أغلب أجهزة الكمبيوتر الحديثة مجموعة أحرف ASCII وتحتوي رموز ASCII علي الأبجدية الإنجليزية وعلامات الترقيم الشائعة وأحرف المسافات.

وفي الولايات المتحدة الأمريكية يستخدم Macs مجموعة أحرف MacRoman وتستخدم أجهزة كمبيوتر Windows مجموعة أحرف تدعي Windows ANSI بينما تستخدم محطات عمل Unix workstations ISO Latin-1. تلك هي كل إمدادات ASCII التي تدعم الأحرف الإضافية مثل Àç المطلوبة في اللغات الأوربية الغربية كالفرنسية والأسبانية وفي مناطق أخري كاليابان واليونان وإسرائيل، تستخدم أجهزة الكمبيوتر مجموعات أحرف عبارة عن خليط من مجموعات الأحرف التي تدعم ASCII بالإضافة إلي اللغة المحلية.

ولكن هذا لا يجدي علي شبكة الإنترنت فلا يمكنك أثناء قراءة أخبار Jose Mercury News San أن تقلب الصفحة لتجد مجموعة أعمدة مكتوبة باللغة الصينية أو الألمانية لكن علي الويب من الممكن أن يتبع المستخدم رابطا لتكون النتيجة تواجد صفحة باللغة الصينية حتى إذا كان المتجول لا يستطيع قراءة الصينية، سيكون من الممتع رؤية إصدار صحيح من اللغة كما هو موضح في "شكل ٧-٥" بدلا من مجموعة أحرف عشوائية كالتي تبدو في شكل ٧-٢.

ويتغلب XML علي هذه المشكلة عن طريق الابتعاد عــن مجموعــات الأحــرف الصغـيرة المحدودة المقتصرة علي بلد بعينها إلي أخرى تشمل كل الكتابات المستخدمة في اللغـــات الحيـــة "وبعض اللغات الميتة أيضا" وتسمى مجموعة الأحرف هذه الشــفرة الموحــدة Unicode وكمـــا ذكرنا أنفا Unicode هي مجموعة أحرف ثنائية البيتات توفر عرضا لأكثر من أربعين ألفاً مـــن الأحرف في مئات الكتابات واللغات ويفهم معــالجو XML Unicode حتـــى إذا لــم يســتطيعوا عرضها بشكل كامل.

وكما عرفت في الفصل السادس مستند XML مقسم إلي كيان نص وكيانات ثنائيـــة ويحتـــوي كيان كل نص علي شفرة وإذا لم يتم تحديد الشفرة بوضوح في تعريف الكيان، فما سيحدث هـــو UTF-8 وهو شكل مضغوط من Unicode وهو يترك نص ASCII الخالص بلا تغيير. ولذلـــك فملفات XML التي لا تحتوي إلا علي أحرف ASCII الشائعة يمكن تحريرها باســتخدام أدوات لا علم لها بتعقيدات التعامل مع مجموعات الحرف الثنائية البينات مثل Unicode.

مجموعة أحرف ASCII

الشفرة الأمريكية القياسية لتبادل المعلومات ASCII هي واحدة من مجموعات الأحرف الأصليـــة والأكثر شيوعا. وهي تمثل قاسما مشتركا لما يجب أن تدعمهم مجموعة الأحرف وهي تعرف كل الأحرف المطلوبة لكتابة اللغة الإنجليزية ولا شئ سواها. ويتم تشغيل الأحرف بأرقام تبـــدأ مـــن صفر وحتى ١٢٧.ويوضح جدول ٧-١ مجموعة أحرف ASCII.

الجدول ٧-١

مجموعة أحرف ASCII

Character	Code	Character	Code	Character	Code	Character	Code
	96	@	64	Space	32	Null(Control-@)	0
A	97	A	65	!	33	start of heading (Control-A)	1
b	98	B	66	"	34	start of text (Control-B)	2
c	99	C	67	#	35	end of text (Control-C)	3
d	100	D	68	$	36	end of transmis- sion (Control-D)	4

الجدول ٧-١

مجموعة أحرف ASCII

Character	Code	Character	Code	Character	Code	Character	Code
e	101	E	69	%	37	Enquiry (Control-E)	5
f	102	F	70	&	38	Acknowledge (Control-F)	6
g	103	G	71	'	39	bell (Control-G)	7
h	104	H	72	(	40	Backspace (Control-H)	8
I	105	I	73	)	41	tab(Control-I)	9
j	106	J	74	(	42	Linefeed (Control-J)	10
k	107	K	75	+	43	vertical tab) (Control-K	11
l	108	L	76	,	44	Formfeed (Control-L)	12
m	109	M	77	-	45	carriage return (Control-M)	13
n	110	N	78	.	46	shift out (Control-N)	14
o	111	O	79	/	47	shift in (Control-O)	15
p	112	P	80	0	48	data link escape (Control-P)	16
q	113	Q	81	1	49	device control 1 (Control-Q)	17
r	114	R	82	2	50	device control 2 (Control-R)	18
s	115	S	83	3	51	device control 3 (Control-S)	19
t	116	T	84	4	52	device control 4 (Control-T)	20
u	117	U	85	5	53	negative acknowl- edge (Control-U)	21
v	118	V	86	6	54	Synchronous idle (Control-V)	22
w	119	W	87	7	55	end of transmission block (Control-	23

الجدول ٧–١

مجموعة أحرف ASCII

Character	Code	Character	Code	Character	Code	Character	Code
						W)	
x	120	X	88	8	56	cancel (Control-X)	24
y	121	Y	89	9	57	end of medium (Control-Y)	25
z	122	Z	90	:	58	Substitute (Control-Z)	26
{	123	[	91	;	59	escape (Control-[)	27
|	124	\	92	<	60	file separator (Control-\)	28
}	125	]	93	=	61	group separator (Control-])	29
~	126	^	94	>	62	record separator (Control-^)	30
delete	127	_	95	?	63	unit separator (Control-_)	31

والأحرف من صفر حتى ٣١ هي أحرف تحكم غير قابلة للطبع. وهـــي تتضمــن أحــرف الإرجاع، تغذية السطر والتبويب والجرس والأحرف المشابهة. والكثير من تلك الأحرف هي مـــا تبقى من أيام أطراف أجهزة إرسال البرقيات المعتمدة على الورق. على سبيل المثــال، تســتخدم أحرف الإرجاع حرفياً بمعنى تحريك الحرف للهامش الأيسر كما تفعل في الآلة الكاتبة. ويحــرك "تغذية السطر" الاسطوانة على السطر.

وأحرف التحكم المذكورة هذه لم تعد تستخدم الآن ومجموعات الأحرف الأخرى التي ستقابلها هي مجموعات أكبر من ASCII فهي تعرف من صفر وحتى ١٢٧ بنفس الطريقة التي تقوم بـــها ASCII. لكنها تضيف أحرف إضافية ابتداء من ١٢٨ فما فوق.

مجموعة أحرف ISO

يمثل حرف A في ASCII الأمريكية وهذا يعني اختصاص ASCII باللغة الإنجليزية وبـــالأخص تلك الخاصة بأمريكا تحديداً ولا تحتوي ASCII على أحرف مثل £ أو ü أو ؟ أو غيرها كـــالتي تحتاجها لكتابة لغات أخرى.

ويمكن مد ASCII بتحديد أحرف إضافية لأرقام تلي رقم ١٢٨. وقـامت منظمـة القياسـات العالمية ISO بتعريف عدد من مجموعات الأحرف المختلفة والمعتمدة على ASCII وهي تصنيف أحرف إضافية لازمة للغات والأماكن الأخرى.ISO 8859-1 والشائع تسميتها Latin-1 وتحتوي Latin-1 على أحرف إضافية لكتابة كل اللغات الأوروبية الغربية. والأحرف من صفر وحتـى ١٢٧ هي نفسها الموجودة في ASCII، أما الأحرف من ١٢٨ وحتى ٢٥٥ فيتـم عرضـها فـي جدول "٧-٢" الاثنان وثلاثون حرفاً الأوائل هم أيضاً غير مستخدمين وهم أحرف تحكم غير قابلة للطبع.

الجدول ٧-٢

مجموعة أحرف ISO 8859-1 Latin-1

Character	Code	Character	Code	Character	Code	Character	Code
	224		192	Non-breaking space	160	Undefined	128
	225		193		161	Undefined	129
	226		194		162	Bph	130
	227		195		163	Nbh	131
	228		196		164	Undefined	132
	229		197		165	Nel	133
	230		198	B	166	Ssa	134
	231		199		167	Esa	135
	232		200		168	Hts	136
	233		201		169	Htj	137
	234		202		170	Vts	138
	235		203	|	171	Pld	139
	236		204	T	172	Plu	140
	237		205	Discretionary hyphen	173	Ri	141
	238		206		174	ss2	142
	239		207		175	ss3	143
	240		208		176	Dcs	144
	241		209		177	pu1	145
	242		210	2	178	pu2	146
	243		211	3	179	Sts	147
	244		212		180	Cch	148

الجدول ٧-٢							
مجموعة أحرف ISO 8859-1 Latin-1							
Character	Code	Character	Code	Character	Code	Character	Code
	245		213		181	Mw	149
	246		214		182	Spa	150
+	247		215	ل	183	Epa	151
	248		216		184	Sos	152
	249		217	1	185	Undefined	153
	250		218	o	186	Sci	154
	251		219	»	187	Csi	155
	252		220	1/4	188	St	156
	253		221	1/2	189	Osc	157
	254		222	3/4	190	Pm	158
	255		223	¿	191	Apc	159

وتفتقر Latin-1 إلى الكثير من الأحرف المفيدة اللازمة لكتابة اللغة اليونانية والسيريلية والصينية والكثير من الكتابات واللغات الأخرى. ربما تعتقد أن مثل تلك اللغات يمكن كتابتها باستخدام أرقام تبدأ من ٢٥٦فما فوق لكن هذه ميزة. يستطيع البايت أن يحمل قيم من صفر وحتى ٢٥٥، لتجاوز هذا الحد يلزم استخدام مجموعة أحرف متعددة البايتات ولأسباب تاريخية، تمت كتابة أغلب البرامج بفرض أن الأحرف والبايتات متماثلتان وأنهما ينفصلان عند التعرض لمجموعات أحرف متعددة البايتات.لذلك، تستخدم معظم أنظمة التشغيل الحالية "باستثناء Windows NT مجموعات أحرف فردية البايتات بدلاً من مجموعة كبيرة متعددة البايتات Latin-1 هي الأكثر شيوعاً من تلك المجموعات لكن تبقى الحاجة إلى مجموعات أخرى يمكنها معالجة اللغات الإضافية".

تقوم ISO 8859 بتعريف مجموعات أحرف أخرى "8859-2 إلى 8859-10 و8859-15" وهي مناسبة لكتابات مختلفة بالإضافة إلى أربع مجموعات أخرى في التطور النشط 8859-11 إلى 8859-14 ويسرد جدول "٧-٣" مجموعات أحرف ISO واللغات والكتابات التي يمكن استعمالها لهم. ويشترك الجميع في أحرف ASCII ذاتها من صفر وحتى ١٢٧ ويحتوي كل مجموعة منهم على أحرف إضافية ابتداء من ١٢٨ وحتى ٢٥٥.

الجدول ٧-٣
مجموعة أحرف ISO

مجموعة الحرف	تعرف بـ	اللغات
ISO 8859-1	Latin-1	ASCII زائد الأحرف اللازمة لمعظم اللغات الأوروبية الغربية مثل الألبانية والأفريقية، الباسك الكتالانية والهولندية والدنماركية والإنجليزية والفنلندية والفاروزية والفلمنكية والفرنسية والألمانية والإيسلندية والأسبانية والسويدية، إلا أنها تحذف الأحرف المزدوجة مثل في الهولندية وŒ في الفرنسية وعلامات التنصيص الألمانية.
ISO 8859-2	Latin-2	ASCII بالإضافة إلى الأحرف اللازمة للغات وسط أوروبا مثل اللغة التشيكية والإنجليزية والألمانية والهنجارية والبولندية والرومانية والكرواتية والسلوفاكية والسلوفيتية والصوبية.
ISO 8859-3	Latin-3	ASCII بالإضافة إلى الأحرف اللازمة للإنجليزية ولغة الأسبرانت والألمانية والمالطية والفرنسية.
ISO 8859-4	Latin-4	ASCII بالإضافة إلى الأحرف اللازمة للغات البلطيق مثل اللاتفية والليتوانية واللمانية والجرين لاندية واللابية وهي تحل محل ISO 8859-10 وlatin-6.
ISO 8859-5		ASCII بالإضافة إلى الأحرف السيريلية اللازمة للغة البايلوروسية والبلغارية والمقدونية والصربية والأوكرانية.
ISO 8859-6		ASCII بالإضافة إلى اللغة العربية.
ISO 8859-7		ASCII بالإضافة إلى اللغة اليونانية.
ISO 8859-8		ASCII بالإضافة إلى اللغة العبرية.
ISO 8859-9	Latin-5	Latin-1 إلا أن الحروف التركية و و و و و تحل محل الأحرف الإيسلندية غير المستخدمة بكثرة مثل و و Ý وÝ.

الجدول ٧-٣
مجموعة أحرف ISO

مجموعة الحرف	تعرف بـ	اللغات
ISO 8859-10	Latin-6	ASCII بالإضافة إلى الأحرف الخاصة باللغات الجيرمانية مثل الليتوانية ولغة الانكيمو اللايسلندية.
ISO 8859-11		ASCII بالإضافة إلى لغة التاي
ISO 8859-12		يمكن استخدامها مع ASCII واللغات الديفاناجارية "الهندية والسانكرتية" ولكن لا يوجد اقتراح حالي.
ISO 8859-13	Latin-7	ASCII بالإضافة إلى حدود البلطيق مثل اللاتفية.
ISO 8859-14	Latin-8	ASCII بالإضافة إلى اللغة الحالية والويلزية.
ISO 8859-15	Latin-9, Latin-0	هي مثل Latin-1 لكن مع وجود علامة يورو بدلاً من العلامة العالمية الحالية كما تحل الحرف الفنلندية Š و Ž و š و ž محل الرموز ¦ و ¨ وتحل الأحرف لفرنسية محل الكسور 1/2, 1/4 ¾

وغالباً ما تتداخل أو تشترك في صفة واحدة هذه المجموعات فيمكن كتابة لغات عديدة كالإنجليزية والألمانية على سبيل المثال، بأكثر من مجموعة أحرف. وإلى حد ما تم تصميم المجموعات المختلفة بصورة تسمح بدمج بعض اللغات المختلفة. على سبيل المثال، تستطيع Latin-1 دمج معظم اللغات الغربية والأيسلندية بينما تدمج Latin-5 معظم اللغات الغربية باللغة التركية بدلاً من الأيسلندية. لذلك، إذا احتجت إلى مستند باللغة الإنجليزية والفرنسية والأيسلندية فعليك استخدام Latin-1. والمستند المتضمن للإنجليزية والفرنسية والتركية أن يستخدم Unicode حيث أنه لا توجد مجموعة أحرف فردية للبايتات يمكنها التعامل مع الثلاث لغات وكتابات.

لا تلائم المجموعة فردية البايتات اللغة اليابانية والصينية والكورية وذلك اللغات لديها أكثر من ٢٦٥ حرف لكل منها، لذلك يجب استخدام مجموعات الأحرف متعددة البايتات.

مجموعة أحرف MacRoman

MacOS تسبق Latin-1 بعدة سنين "حيث أن مقياس ISO 8859-1 تم استخدامه في عام ١٩٨٧ بينما صدرت أول Mac في عام ١٩٨٤". وهذا يعني أن شركة آبل كان عليها أن تعرف مجموعة الأحرف الموسعة الخاصة بها والتي تدعى MacRoman لديها مجموعة الأحرف الموسعة ذاتها

مثل Latin-1 باستثناء الأحرف الأيسلندية T وy وe لكن تم تحديد الحـــــروف بأرقـــام مختلفـــة MacRoman تماثل ASCII وLatin-1 في الشفرات الخاصة بالـ ١٢٧ حرفاً الأوائل وهذا مـــن ضمن الأسباب التي تستخدم الأحرف الموسعة تبدو مضحكة عندما تنتقل من PC إلـــــى Mac أو العكس ويسرد جدول "٧-٤" النصف الأعلى من مجموعة أحرف MacRoman.

الجدول ٧-٤

مجموعة أحرف MacRoman

Character	Code	Character	Code	Character	Code	Character	Code
‡	224	¿	192		160		128
·	225	i	193	¢	161		129
'	226		194	£	162		130
"	227	√	195	§	163		131
‰	228	ƒ	196	•	164		132
	229		197	¶	165		133
	230	Δ	198	ß	166		134
	231	«	199	®	167		135
	232	»	200	©	168		136
	233		201	™	169		137
Î	234	Non breaking space	202	´	170		138
Ï	235		203	¨	171		139
Ï	236		204	≠	172		140
Ì	237	Õ	205	Æ	173		141
Î	238	Œ	206	Ø	174		142
Ô	239	Œ	207	∞	175		143
Ö	240	–	208	±	176		144
Apple	241	—	209	≤	177		145
Ó	242	"	210	≥	178		146
Ü	243	"	211	¥	179		147
Ü	244	'	212	¥	180		148

الجدول ٧-٤

مجموعة أحرف MacRoman

Character	Code	Character	Code	Character	Code	Character	Code
ı	245		213	µ	181		149
ˆ	246		214	∂	182		150
˜	247		215	Σ	183		151
¯	248	÷	216	∏	184		152
˘	249	◊	217	π	185		153
˙	250	Ÿ	218	∫	186		154
˚	251	Ÿ	219	ª	187		155
¸	252	/	220	º	188		156
˝	253	‹	221	Ω	189		157
˛	254	›	222	Æ	190		158
ˇ	255	fi	223	ø	191		159

مجموعة أحرف Windows ANSI

الإصدار الأول الذي أصدرته Windows وحصل على إقرار عريض قد تلي Mac ببضعة سنوات لذلك كان باستطاعته اختيار مجموعة أحرف Latin-1. وهو قد أبدل أحرف التحكم غير قابلة للطبع والتي بين ١٣٠ و١٥٩ بأحرف طباعة أكثر وذلك لإطالة المدى المتاح بعض شيء. هذا الإصدار المعدل لـ Latin-1 يدعى Windows ANSI ويسرد جدول "٧-٥" أحرف Windows ANSI.

الجدول ٧-٥

مجموعة أحرف الشفرة الموحدة

Character	Code	Character	Code	Character	Code	Character	Code
˜	152	Undefined	144		136	Undefined	128
™	153	[illegible]	145	ن	137	Undefined	129
[illegible]	154	‰	146		138	'	130

الجدول ٧-٥

مجموعة أحرف الشفرة الموحدة

Character	Code	Character	Code	Character	Code	Character	Code
›	155	Š	147		139	□	131
Œ	156	‹	148		140	„	132
Undefined	157	Œ	149	Undefined	141	†	133
Undefined	158		150	Undefined	142		134
Ÿ	159		151	Undefined	143	‡	135

مجموعة أحرف الشفرة الموحدة Unicode

استخدام مجموعات أحرف مختلفة لكتابات ولغات متعددة يعمل بشكل جيد بشرط:

١-عدم الحاجة إلى العمل بأكثر من كتابة في وقت واحد.

٢-عدم تبادل الملفات مع أي شخص باستخدام مجموعة أحرف مختلفة.

وحيث أن Macs وPCs يستخدما مجموعات أحرف مختلفة، فالكثير من الناس لا ينجحوا في التعامل بهذا المقياس وأصبحت الحاجة ضرورية إلى مجموعة أحرف واحدة يتفق عليها الجميع وتحمل شفرة لكل الأحرف الموجودة في كل كتابات العالم. وإنشاء مثل ذلك تلك المجموعة هو أمر صعب لأنه يتطلب فهماً مفصلاً لمئات اللغات وطرق كتابتها والأصعب هو جعل مطوري البرامج يتفقون على استخدامها بمجرد إنشائها. والعمل قائم الآن على إنشاء مجموعة كهذه وتسمى الشفرة الموحدة Unicode والمسوقين الأساسيين مثل Microsoft وApple وIBM وSun وBe وغيرهم في طريقهم للتعامل معها.

ويقوم XML بتخصيص Unicode بتشفير كل حرف على هيئة حرف ثنائي البايتات غير محدد بعلامة وله قيمة تتراوح بين صفر و٥٣٥ و٦٥ وحالياً يتم تعريف أكثر من ٤٠,٠٠٠ من أحرف Unicode المختلفة أما ٢٥,٠٠٠ مكاناً الباقين فهم محجوزين لأية توسعات مستقبلية. ويستخدم حوالي عشرين ألفاً من الأحرف لرموز ايديوغرافات هان ويستخدم أحد عشرا تقريباً للمقاطع اللفظية الخاصة باللغة الكورية الهانجولية أما بقية الأحرف فهي تشفر معظم ما تبقى من لغات العالم وتتطابق أحرف Unicode من صفر وحتى ٢٥٥ مع أحرف Latin-1 ابتداء أيضاً من صفر وحتى ٢٥٥.

ولقد كنت أرغب في أن أعرض لكم جدولا أحرف Unicode، لكن للأسف إن فعلت ذلك فلن يحتوي الكتاب إلا على هذا الجدول فقط وللاطلاع على مزيد من الشفرات المحددة للأحرف المختلفة في Unicode عليك بالحصول على نسخة من كتاب The Unicode Standard "الطبعة الثانية ISBN 0-201-48346-9 من Addison-Wesley". ويحتوي هذا الكتاب ذو التسع وخمسين صفحة على تحديدات Unicode 2.0 الكاملة. بما فيها من تخطيطات أحرف لجميع الأحرف المختلفة المعرفة في Unicode 2.0. يمكنك العثور على معلومات فورية على موقع الويب الخاص Unicode Consortium في http://www. unicode و/ http://charts.unicode.or ويقوم جدول "٧-٦" برد الكتابات المختلفة المنفردة في Unicode والتي تعرفك بتعددية Unicode في الاستعمالات ويتم تشفير الأحرف في كل كتابة عن طريق نطاق فرعي "كتلة" مترابط من نقاط الشفرات البالغ عددهم ٦٥,٥٣٦ في Unicode ويمكن كتابة معظم اللغات باستخدام الأحرف في أي من تلك الكتل فمثلاً يمكن كتابة اللغة الروسية باستخدامها الكتلة السريالية إلا أن بعض اللغات مثل الكرواتية أو التركية تحتاج لمزج وتوفيق أحرف من الكتل اللاتينية الأربعة الأولى.

الجدول ٧-٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Basic Latin	0-127	ASCII والإنجليزي الأمريكي
Latin-1 Supplement	126-255	النصف العلوي من ISO Latin-1 بانضمامه إلى كتلة Basic Latin يستطيع التعامل مع اللغة الدانمركية والهولندية والإنجليزية والفاروزية والفلمنكية والألمانية ولغة هاواي والأيستلندية والأندونيسية والأيرلندية والإيطالية والنروجية والبرتغالية والأسبانية والسواصلي والسويدية.

الجدول ٧-٦
كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Latin Extended-A	256-383	تضيف هذه الكتل الأحرف من مجموعات ISO 8859 وهي Latin-2 وLatin-3 وlatin-4 وLatin-5 والتي ليست موجودة في كتل Basic Latin وLatin-1 والترابط بتلك الكتل تستطيع هذه الكتلة أن تشفر الأفريكان والبريتونية والباسك والكاتالانية والتشيكية والأسبرانتو والاستوانية والفرنسية والفرويريانية لغة جرين لاند والهانجارية واللاتفانية والليتوانية والمالطية والبولندية والبروتيال والراتيوررومانية ولغة رومانيا والسلوفاك والسلوفينية والصربية والتركية والويلزية.
Latin Extended-B	383-591	أغلب الأحرف المطلوبة لمد الكتابة اللاتينية لتستطيع التعامل مع اللغات غير المكتوبة تقليدياً في هذه الكتابة وهي تتضمن الكثير من اللغات الأفريقية والحرفين الذين يمثلان صوتاً مفرداً كما في اللغة الكرواتية لتلائم الحروف الصربية السيريلية والرموز الصوتية Pinyin اللغة الصينية والحرف السامية من Latin-10.
IPA Extensions	592-687	الأبجدية الصوتية العالمية
Spacing Modifier	686-767	الرموز الصغيرة التي تغير الحرف صوتياً.
Combining Diacritical	766-879	سيتم عن طريقها دمج العلامات الصوتية المميزة التي توضع فوق أحرف أو تحته مثل ~ و' و مع الحرف السابق ويتم وضعها أعلى الحرف بدلاً من رسمها كحرف منفصل.

الجدول ٧-٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
اليونانية	880-1023	اليونانية الحديثة المعتمدة على ISO 8859-7 كما توفر الأحرف اللغة القبطية أيضاً.
السيريلية	1024-1279	الروسية واللغات السلافية الأخرى مثل الأوكرانية والبيلوروسية وغيرها وبعض اللغات غير السلافية الخاصة بالاتحاد السوفيتي السابق مثل الأذربيجانية والأوزبكستانية والكباردية والشيشانية والطاجاكية وما إلى ذلك والمعتمدة على ISO 8859-5 بعض اللغات مثل الكردية والأبخازية تتطلب كلاً من الحرف اللاتينية والسيريلية.
Armenian	1326-1423	الأرمنية
Hebrew	1424-1535	العبرية القديمة والحديثة مثل اليديشية واليوديزمية والآرامية.
Arabic	1536-1791	العربية والفارسية والباشتو (الإيرانية) ولغة ايستند والكردية والتركية القديمة.
Devanagari	2304-2431	السانسكريتية والهندية وبقية لغات شبه القارة الهندية مثل المراثية والنيبالية والبهاشبورية والبهيلية والسنسراجي بالاهاجية والمشامابيسجارية والجارهوالية والجوندية والهاريانية ولغة الهو والجايبوري والكولو والكوماوني والكوركر والكوروك والمادوراي والموندري النيواري والباليا والماثالي.

الجدول ٧-٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Bengali	2432-2559	وهي كتابة لشمال الهند تستخدم في ولاية البنغال في غرب الهند وبنجلادش. وهي تستخدم للغة البنغالية والآساميدية والدفلة والجارو والهالام والخامس والمانيبوري، والميزو والتاجا والموندا والرايان السانتالي.
Gurmukhi	2560-2687	البانجابية
Gujarati	2686-2815	الجوجاراتية
Oriya	2816-2943	أوريا والكوندي والسانتالي
Tamil	2944-3071	التاميل والباداجا المستخدمة في جنوب الهند وسريلانكا وسنغافورة وأجزاء من ماليزيا
Telugu	3072-3199	تيلوجو وجوندي ولامبادي
Kannada	3200-3327	كانادا وتولو
Malalayam	3326-3455	مالالايام
Thai	3584-3711	تاي وكيوولافانا وبالي
Lao	3712-3839	لاو
Tibetan	3840-4031	لغات الهيمالايا بما فيها لغة التبت ولاداكي ولاهولي
Georgian	4256-4351	الجورجية وهي لغة جمهورية جورجيا الروسية السابقة على البحر الأسود
Hangul Jamo	4352-4607	المكونات البجدية لألفاظ اللغة الكورية الهانغولية
Latin Extended	7680-7935	الأحرف اللاتينية العادية مثل ٤ و٣ لكن مع دمجها بالعلامات الصوتية التي يندر استخدامها إلا مع الأحرف المتحركة الخاصة باللغة الفيتنامية

الجدول ٧-٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Greek Extended	7936-8191	الأحرف اليونانية مدمجة مع العلامات الصينية المستخدمة في اليونانية القديمة والبوليتونك
General Punctuation	8192-8303	علامات الترقيم المصنفة
Superscripts and	8304-8351	الأحرف الفوقية والتحتية الشائعة
Currency Symbols	8352-8399	رموز العملات غير الموجودة في الكتل الأخرى
Combining Marks for	8400-8447	تستخدم لعمل امتداد علامات للعلامات الصوتية لحرفين أو أكثر
Letter like Symbols	8446-8527	وهي الرموز التي تبدو مثل الأحرف مثل و—
Number Forms	8526-8591	للكسور والأرقام الرومانية
Arrows	8592-8703	للأسهم
Mathematical	8704-8959	عوامل التشغيل الحسابية التي تظهر في الكتل الأخرى
Miscellaneous	8960-9039	علامات الاقتصاص ورموز الأقواس من ميكانيكا الكم والرموز اللازمة للغة برمجة API والرموز الفنية المنسقة
Control Pictures	9216-9279	صور أحرف تحكم ASCII والتي تستخدم عامة في اكتشاف التصحيح ومجموعات الشبكات
Optical Character	9280-9311	OCR-A وMICR "تعرف على حرف الحبر المغناطيسي" والرموز على الشبكات المطبوعة

الجدول ٧–٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Enclosed	9312-9471	الأحرف والأرقام في الدوائر والأقواس
Box Drawing	9472-9599	الأحرف لرسم المربعات على محطات طرفية أحادية المسافات
Block Elements	9600-9631	رسوم محطات طرفيـة أحاديـة المسـافات كالمستخدمة في DOS وغيرها
Geometric Shapes	9632-9727	المربعات والمثلثات والمعينات وغيرها
Miscellaneous	9726-9983	كروت ورق اللعب والشطرنج ولذلك وما إلى ذلك
Dingbats	9984-10175	أحرف Zapf Dingbat
Punctuation	12286-12351	الرموز وعلامات الترقيم المستخدمة في اللغة الصينية واليابانية والكورية
Hiragana	12352-12447	أبجدية مقطعية مكتوبة بأحرف مطبعية متصلة وهي خاصة باللغة اليابانية
Katakana	12446-12543	أبجدية مقطعية غير مكتوبة بأحرف مطبعيـة متصلـة وهـي تسـتخدم لكتابـة الكلمـات المستوردة من الغرب إلى اللغة اليابانية علـى الأخص الكلمات الحديثة مثل (لوح المفـاتيح) أبجدية صوتية للغة الصينية وهـي تسـتخدم أساساً للتعليم
Bopomofo	12544-12591	أبجدية صوتية للغة الصينية وهـي تسـتخدم أساساً للتعليم
Jamo	12687	الأحرف الكورية اللازمة للتناسق مع شـفرة KSC 5601
Kanbun	12686-12703	العلامات المستخدمة فـي اليابانيـة لتحديـد ترتيب القراءة للصينية القديمة

الجدول ٧–٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
CJK Enclosed Letters	12800-13055	أحرف هانجول وكاتاكانا مرفقة داخل دوائر وأقواس
CJK Compatibility	13056-13311	الأحرف اللازمة فقط لتشفير KSC 5601 وCNS 11643
CJK Unified	19966-40959	رموز أيديوغراف هان المستخدمة للغة اليابانية والصينية والكورية
Hangul Syllables	44032-55203	أبجدية مقطعية كورية
Surrogates	55296-57343	غير مستخدمة حاليا لكنها ستسمح باتساع Unicode لأكثر من مليون حرف مختلف
Private Use	57344-63743	يمكن لمطوري البرامج أن يضمنوا أحرفهم المخصصة هنا وهي غير متوافقة خلال التنفيذ
CJK Compatibility	63744-64255	رموز أيديوغراف إضافية تلزم فقط للحصول على توافق مع المقاييس الموجودة مثل KSC 5601
Alphabetic Presen-	64256-64335	الأحرف المزدوجة والتهجيات المختلفة المستخدمة احياناً في اللاتينية والأرمنية والعبرية
Arabic Presentation	64336-65023	تهجيات مختلفة من الأحرف العربية المصنفة
Combining Half Marks	65056-65071	دمج علامات صوتية مختلفة في علامة صوتية واحدة يمكنها أن تسع أحرف كثيرة
CJK Compatibility Forms	65072-65103	تهجيات رأسية مختلفة لكلمة من أيديوغرافات هان المستخدمة في تايوان

الجدول ٧-٦

كتل كتابة Unicode

شكل الخط	النطاق	الاختصاص
Small Form Variants	65104-65135	إصدار مصغر من علامات ترقيم ASCII المستخدمة أساساً في تايوان
Additional Arabic Presentation Forms	65136-65279	تهجيات مختلفة للأحرف العربية المصنفة
Half-width and Full- width Forms	65280-65519	أحرف تساعد على التحول من شفرات مختلفة للغة الصينية واليابانية خاصة بنفس الأحرف
Specials	65520-65535	علامة ترتيب بايت وعرض الصفر ولا توجد علامة فاصلة لبدء ملفات Unicode

UTF-8

حيث أن Unicode يستخدم إثنتين من البيتات لكل حرف،فملفات النصوص الإنجليزية حجمها مضاعف في Unicode بما هو الحال في ASCII وUTF-8 Latin-1 هي إصدار مضغوط من Unicode character وتستعمل بايت واحد للأحرف الشائعة مثل أحرف ASCII التي تبدأ من صفر وحتى ١٢٧ مع الحاجة لاستعمال ثلاثة بايتات للأحرف غير الشائعة مثل مقاطع الألفاظ الهانغولية ورموز ايديوغراف هان. إذا كنت تكتب بالإنجليزية في أغلب الأوقات UTF-8 يمكنها تقليل حجم الملف بأكثر من خمسين بالمائة أما إذا كنت تكتب بالصينية أو الكورية أو اليابانية UTF-8 يمكنها مضاعف حجم الملف بأكثر من خمسين بالمائة لذلك يجب توخي الحذر عند استخدامها. ولا تؤثر UTF-8 على الكتابات غير الرومانية وغير CJK مثل اليونانية والعربية والسيريلية والعبرية.

وتفترض معالجاتXML أن معلومات النص في تنسيق UTF-8 حتى يثبت العكس. وهذا يعني أن باستطاعتها قراءة ملفات ASCII لكن باقي التنسيقات مثل MacRoman وLatin-1 تسبب لها مشاكل

نظام الأحرف العالمي

يوجه النقد للشفرة المحلية باعتبار أنها لا تحتوي علي ما يكفي من اللغات على الأخص بما يتعلق بلغات شرق أسيا وهي تعرف عشرين ألفا من الثمانين آلف من رموز هان المستخدمة في

الصينية واليابانية والكورية والفيتنامية التاريخية "حيث تستخدم نظام الفيتنامية الحديثة الأبجدية الرومانية".

يستخدم نظام الأحرف "٣١بيت" يسمح لأكثر من اثنين مليون حرف مختلفة. وهذا يعطي كل حرف مستخدم في أية لغة علي وجه الأرض كما يساعد UCS علي تحديد مجموعة أحرف كاملة لكل لغة بحيث يكون حرف (e) في الفرنسية مختلفا عن شبيهة في الإنجليزية أو الألمانية.

وتماما مثل Unicode، يعرف UCS عددا مختلفا من طرق الهجاء للكلمات والهيئات المضغوطة.تتم الإشارة للشفرة الموحدة الخالصة UCS ثنائية الباتات. UTF-16 هي شفرة خاصة تخطط بعض أحرف UCS في سلاسل بايتات مختلفة الأطوال بحيث لا تتغير معلومات UCS-2 Unicode.

ميزات UCS التي تتفوق فيها علي Unicode ما هي إلا عملية نظرية. والأحرف التي تم تعريفها فعليا في UCS هي نفسها الموجودة بالفعل في الشفرة المحلية. لكنها توفر مكانا اتساع في المستقبل.

كيفية كتابة XML في الشفرة المحلية

الشفرة الموحدة الشفرة المحلية هي مجموعة الحروف الأصلية فيXML وسيعرضها مستعرضو XML بصورة جيدة علي الأقل بالخطوط المتاحة لكن الخطوط لا تتوافر لكثير إذا كان محررو النص لا يدعمه النطاق الكامل من الشفرة الموحدة.

يجب عليك معالجة هذه المشكلة بأي من الوسيلتين التاليتين.

١-اكتب باستخدام مجموعة أحرف محلية مثل Latin-3 والتي تحول ملف إلى Unicode.

٢-ضمن مراجع أحرف Unicode character في النص الذي يتعرف رقميا علي أحرف محددة.

الاختيار الأول مفضل عندما يكون لديك كمية هائلة من النصوص لإدخالها في كتابة واحدة بالضرورة أو في كتابة واحدة بالإضافة إلى ASCII. ويكون الاختيار الثاني مفيدا عندما تكون في حاجة إلى خلط أجزاء صغيرة من كتابات متعددة داخل المستند.

إدراج الأحرف في ملفات XML باستخدام مراجع الأحرف

كل حرف Unicode هو عبارة رقم بين ٠ و٦٥،٥٣٥ إذا لم يكن لديك محرر نص يمكنه الكتابة في Unicode، يمكنك استخدام مرجع للحرف لإدراج الحرف في ملف XML بدلا من ذلك.

يتكون مرجع Unicode من الحرفين &# وتتبعهم شفرة الحرف ثم (؛) فاصلة منقوطة على سبيل المثال، الحرف اليوناني π له قيمة في Unicode وهي ٩٦٠ لذلك يمكن إدخاله في ملف XML علي شكل π والحرف السريالي Ҷ له قيمة في Unicode تساوي ١٢٠٦ لذلك يمكن تضمينه في ملف XML بمرجع Ҷ.

يمكن تخصيص مراجع أحرف Unicode في ست عشري "قاعدة ١٦" رغم أن معظم الناس تر تاج في التعامل بالأرقام العشرية، تعطي تخصيصات Unicode قيم للأحرف علي شكل أرقام ست عشرية ثنائية الباتات. أحيانا يكون استخدام القيم الستة أسهل من تحويلها إلى عشرية. كل ما عليك فعله هو إضافة x بعد &# لتدل علي أنك تستخدم قيمة ست عشرية.

علي سبيل المثال π له قيمة ست عشرية 3C لذلك يمكن إدخاله في ملف XML على شكل π. والحرف السريالي Ҷ قيمته الست عشرية هي 4B6 لذلك يمكن إدخاله في ملف XML علي شكل Ҷ. وإذا كانت الباتتان تنتجان أربعة أرقام ست عشرية عدد هم أربعة أرقام.

يمكن استخدام مراجع أحرف الشفرة الموحدة الست عشرية والعشرية لتضمنين أحرف يمكن أن تترجم في غير هذه الحالة على أنها علامات تحديد على سبيل المثال، تم تشفير علامة الضم (&)في رقم &و &،..وعلامة (<) تشفر على شكل <و< .

التحويل من وإلى الشفرة المحلية

تعالج برامج التطبيق التي تصدر ملفات XML مثل Adobe Framemaker التحويل إلى Unicode أو UTF-8 تلقائيا وإلا ستحتاج إلى استخدام أداة تحويل. وتحتوي "JDK" "مجموعة تطوير جافا" المتوافرة مجانيا علي أداة مساعدة بسيطة خاصة بسطر الأوامر تسمي native2ascii وهي تقوم بالتحويل بين مجموعات الأحرف المحلية الشائعة وغير الشائعة وبين الشفرة الموحدة. على سبيل المثال، يحول الأمر التالي ملف نص يدعي myfile.txt من الشفرة الفرضية للنظام الأساسي إلى Unicode.

```
C:\> native ascii myfile.txt myfile.uni
```

يمكنك تخصيص شفرات أخرى باستخدام خيار encodin–:

```
C:> native2ascii -encoding Big5 chinese.txt chinese.uni
```

كما يمكنك عكس العملية للذهاب من الشفرة الموحدة إلى شفرة محلية باستخدام خيار reverse.

```
C:> native2ascii -encoding Big5 -reverse chinese.uni chinese.txt
```

إذا ترك اسم ملف الإخراج سيتم طبع الملف المحول.

يعالج برنامج native2ascii هروب شفرة نمط جافا الموحدة. وهو عبارة عن أحرف مضمنة علي هيئة u09E3\ وهي ليست بنفس التنسيق كما في مراجع الأحرف الرقمية في XML رغـــم تشابهها. إذا حولت إلى الشفرة الموحدة باستخدام native2ascii فيكون بإمكانك استخدام مراجــع أحرف XML حيث سيستطيع المشاهد التعرف عليهم.

كيفية كتابة XML في مجموعات أحرف أخرى

إن لم يتم الأخبار بغير ذلك، يعتبر معالج XML أن أحرف وجود النص قد تم تشفيرها فـــي UTF-8. وحيث أن UTF-8 تحتوي علي ASCII بسهولة عن طريق معالجي XML.

ومجموعة الحرف الوحيدة غير UTF-8 التي يتطلب فهمها عن طريـــق معـــالج XML هـــي الصرفة. إذا لم يكن باستطاعتك تحويل النص إلى UTF-8 أو صف Unicode يمكنك ترك النص في مجموعة الحرف الأصلية ثم أخبار معالج XML بالمجموعة التي تنتمي إليها. ويجب أن يكون هذا هو الملجأ الوحيد لأنة لا يوجد ضمان أن معالج XML التحكمى يمكنـــه معالجـــة الشـــفرات الأخرى تســـتطيع Nonetheless Netscape Navigator و Internet Explorer تفســـير مجموعات الأحرف الشائعة.

لتحذير معالج XML أنك تستعمل شفرة ليست تابعة للشفرة المحلية يجـــب أن تضمـــن ســـمة encoding في إعلان XML في بداية الملف. على سبيل المثال، لتحديد إن الملف بالكامل يستخدم فرضياً Latin-1 إلا إذا تم تجاوز بتوجيه معالج في كينونة مضمنة عليك باستخدام إعـــلان LMX التالي:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

يمكنك أيضاً تضمين إعلان الشفرة كجزء من أمر معالجة منفصل بعد إعلان XML لكن قبـــل ظهور أي معلومة عن الحرف

```
<?xml encoding='ISO-8859-1"?>
```

يسرد جدول "٧-٧" الأسماء الرسمية لمجموعات الأحرف الشائع استخدامها في الوقت الحـــاضر كما يتم عرضهم في سمات شفرة XML. راجع القائمـــة الرســـمية التـــي توفرهـــا Internet Assigned Numbers Authority للعثور على الشفرات غير الموجودة في هذه القائمة وموقـــع (IANA) هو .http://www.isi.edu/in-notes/iana/assignments/character-sets

الجدول ٧-٧

أسماء مجموعات الأحرف المشتركة

Character Set Name	Languages/Countries
US-ASCII	English
UTF-8	Compressed Unicode
UTF-16	Compressed UCS
ISO-10646-UCS-2	Raw Unicode
ISO-10646-UCS-4	Raw UCS
Character Set Name	Languages/Countries
ISO-8859-1	Latin-1, Western Europe
ISO-8859-2	Latin-2, Eastern Europe
ISO-8859-3	Latin-3, Southern Europe
ISO-8859-4	Latin-4, Northern Europe
ISO-8859-5	ASCII plus Cyrillic
ISO-8859-6	ASCII plus Arabic
ISO-8859-7	ASCII plus Greek
ISO-8859-8	ASCII plus Hebrew
ISO-8859-9	Latin-5, Turkish
ISO-8859-10	Latin-6, ASCII plus the Nordic languages
ISO-8859-11	ASCII plus Thai
ISO-8859-13	Latin-7, ASCII plus the Baltic Rim languages,. particularly Latvian
ISO-8859-14	Latin-8, ASCII plus Gaelic and Welsh
ISO-8859-15	Latin-9, Latin-0; Western Europe
ISO-2022-JP	Japanese
Shift_JIS	Japanese, Windows
EUC-JP	Japanese, Unix
Big5	Chinese, Taiwan
GB2312	Chinese, mainland China
KOI6-R	Russian
ISO-2022-KR	Korean
EUC-KR	Korean, Unix
ISO-2022-CN	Chinese

خلاصة

تعلمت في هذا الفصل:

- يجب أن تتعرف صفحات الويب على الشفرة التي تستخدمها.
- ما هو البرنامج الصغير وما علاقته باللغات والأربعة أشياء التي يتطلبها البرنامج
- كيفية استخدام البرامج الصغيرة في الحاسب عن طريق مجموعات الأحرف والخطـــوط وصور الرموز وطرق الإدخال.
- الاستخدام الشائع لمجموعات الأحرف في نظم أساسية مختلفة واعتماد معظمـــها علـــى ASCII.
- كيفية كتابة XML في Unicode دون وجود محرر Unicode "اكتـــب المســـتند فـــي ASCII ثم ضمن مراجع أحرف Unicode".
- عند كتابة XML بشفرات أخري، ضمن سمة encodings في إعلان XML.

في الفصل التالي، ستكتشف DTDs وكيف يساعدك على تعريف وتقويـــة الكلمـــات وبنـــاء الكلمات وبناء الجملة والقواعد النحوية للمستند.

الجزء

٢

في هذا الجزء

الفصل ٨

تعريفات نوع المستند والتحقق

من صحتها

الفصل ٩

الكيانات والمراجع الكيانية

ومجموعات نوع المستند

الخارجية

الفصل ١٠

إعلان السمات في DTD

الفصل ١١

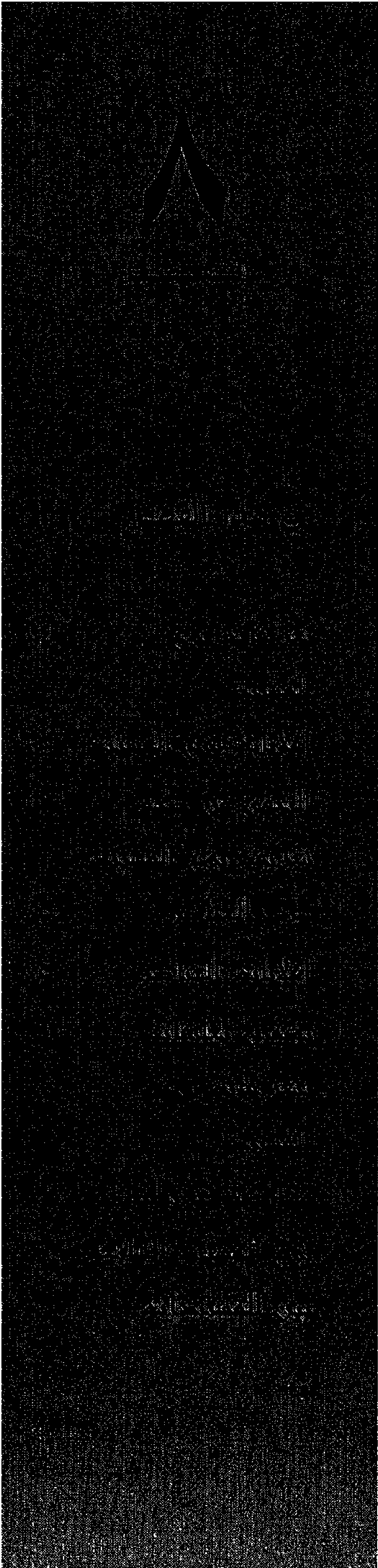
إضافة بيانات التنسيق إلى

XML

التعليقات الختامية والنص غير

الرسومي

تعريفات نوع المستند



تعريفات نوع المستند والتحقق من صحتها

يُعتبر XML إحدى لغات الترميز التفصيلية، أي أنها لغة لوصف لغات الترميز، وسوف تتعلم في هذا الفصل كيفية دعم لغات الترميز الجديدة، والتي تقوم بإنشائها، بالوثائق ووصفها. ويتم تعريف مثل هذه اللغات "والمعروفة أيضاً بمجموعات العلامات tag sets" عن طريق تعريف نوع المستند "DTD" وهو ما يتناوله هذا الفصل بالتفصيل. ويمكن مقارنة المستندات المستقلة بتعريف نوع المستند DTD. وذلك فيما يعرف بعملية التحقق من الصحة. وإذا طابق المستند القيود المدرجة في DTD، فسوف يكون المستند سليماً. وإذا لم يطابقها فسوف يكون غير صالح.

تعريفات نوع المستند

ترمز التسمية المختصرة DTD إلى document type definition أو تعريف نوع المستند. ويقوم تعريف نوع المستند بعرض قائمة بالعناصر وسمات ومجموعات الرموز والكينونات التي يحتويها المستند، بالإضافة إلى علاقة كل منها بالآخر. وتقوم DTDs بتعيين مجموعة من القواعد الخاصة ببنية المستند. فعلى سبيل المثال، قد يشير DTD إلى أن عنصر BOOK له تابع ISBN واحد فقط، وتابع TITLE واحد فقط وتابع AUTHOR أو أكثر وقد أو قد لا يحتوي على SUBTITLE واحد. ويقوم DTD بذلك بواسطة قائمة بإعلانات الترميز لعناصر وكينونات وسمات ومجموعات رموز معينة.



يتناول هذا الفصل إعلانات العناصر بالتفصيل. وتعرض الفصول ٩ و١٠ و١١ للكينونات ومجموعات الرموز على التوالي.

ويمكن تضمين DTDs داخل ملف يحتوي على المستند الذي تقوم بوصفه، أو يمكن ربطها من أحد URL الخارجية. ويمكن إشراك مثل DTDs الخارجية هذه في مستندات ومواقع ويب مختلفة. وتقدم DTDs وسائل للتطبيقات والتنظيمات والمجموعات الهامة تساعد على ملاءمتها معاً ودعمها بالمستندات وتشديد الالتزام بمعايير الترميز.

فعلى سبيل المثال، قد يرغب الناشر في أن يقوم أحد الكتاب بتنفيذ تنسيق معين لجعل تخطيط الكتاب أكثر سهولة. وقد يفضل كاتب آخر كتابة الكلمات في صف دون الاهتمام بمطابقة كل نقطة تعداد رقمي أمام الفصل مع رأس فرعية داخل الفصل. وإذا كان الكاتب يكتب في XML فسوف يكون من السهل للناشر التحقق مما إذا كان الكاتب يلتزم بتنسيق تم تعيينه مسبقاً بواسطة DTD، ومعرفة موضع وكيفية انحراف الكاتب عن التنسيق. ويعتبر ذلك أسهل كثيراً من أن يقوم المحررون بقراءة المستند كله آملين اكتشاف الانحرافات الثانوية عن التنسيق على أساس النمط وحده.

وتساعد DTDs أيضاً في أنها تضمن إمكانية قراءة الأشخاص والبرامج للملفات الأخرى. فعلى سبيل المثال، إذا اتفق الكيميائيون على أحد DTD الخاص بمجموعات الرموز الكيميائية الأساسية، والمتاحة بواسطة تنظيم مناسب عالي الجودة مثل American Chemical Society فلهم أن يطمئنوا إلى أنه بإمكانهم جميعاً قراءة وفهم أوراق الآخرين. ويقوم DTD بتعريف ما هو متاح ظهوره داخل المستند وما هو غير مسموح بظهوره، وذلك بصورة دقيقة.

ويقوم DTD بإقامة معيار للعناصر التي يجب أن تقوم برامج العرض والتحرير بدعمها. والأهم من ذلك أنه يقوم بتثبيت ملحقات غير التي يعلن DTD عدم صلاحيتها. لذلك فهو يمنع بائعي البرامج من تضمين وتوزيع بروتوكولات مفتوحة لحصر المستخدمين داخل البرامج المملوكة لهم.

وبالإضافة إلى ذلك، فإن DTD يعرض عناصر مختلفة للصفحة مرتبة دون تقديم البيانات الخاصة بها فعلياً. ويمكنك DTD من رؤية بنية مستندك منفصلة عن البيانات الفعلية. ويعني ذلك إمكانية وضع العديد من الأنماط والتنسيقات الجميلة داخل البنية الضمنية أو الأساسية دون تشويهها، تماماً كما لو كنت تقوم برسم منزل دون تغيير خريطته المعمارية. وقد لا يتمكن القارئ لصفحاتك من عرض أو حتى معرفة البنية الضمنية، إلا أنه طالما تواجدت هذه البنية، فيمكن أن يستخدمها كل من المبدعين وJavaScripts وCGIs وقواعد البيانات والبرامج الأخرى.

وهناك ما هو أكثر من ذلك والذي تستطيع القيام به عن طريق DTDs. فيمكنك استخدامها لتعريف كينونات ملحق بالكلمات يقوم بإدراج نص متداول مثل مجموعة إرشادات أو عنوان ويتيح لك التزام من يقوموا بإدخال البيانات بالتنسيق الذي تريده. ويمكنك نقل البيانات إلى ومن قواعد البيانات العلائقية والعينية. ويمكنك حتى استخدام XML كتنسيق وسيط لتحويل تنسيقات مختلفة باستخدام DTDs مناسبة. ولنبدأ الآن لنرى كيف تبدو DTDs بالفعل.

إعلانات نوع المستند

يقوم إعلان نوع المستند أو document type declaration بتحديد DTD الذي يستخدمه المستند. ويظهر إعلان نوع المستند في التسجيل الأولى للمستند بعد إعلان XML ولكن قبل العنصر الجذري. وقد يحتوي على تعريف نوع المستند أو على URL الذي يقوم بتعريف الملف حيث تم إيجاد تعريف نوع المستند. وقد يحتوي حتى على كل من المجموعات الفرعية الداخلية والخارجية وذلك في الحالة التي يكون لتعريف نوع المستند فيها قسمان.

تنبيه

لا يمكن اعتبار "إعلان" نوع المستند هو نفسه "تعريف" نوع المستند. فإن تعريف نوع المستند هو فقط ما يشار إليه بالاختصار DTD. ولابد أن يحتوي إعلان نوع المستند، أو يشير إلى، تعريف نوع المستند، إلا أن تعريف نوع المستند لا يحتوي أبداً على إعلان نوع المستند. وقد لا يصنع ذلك فارقاً كبيراً. ففي أغلب الأحيان لا يمثل الفارق بين الاثنين أمراً هاماً.

ويمكنك استدعاء تعليمات البرمجة ٣-٢ "greeting.xml" من الفصل الثالث. وهي موضحة أسفل:

```
<?xml version="1.0" standalone="yes"?>
<GREETING>
Hello XML!
</GREETING>
```

ويحتوي هذا المستند على عنصر واحد وهو GREETING "تذكر أن: <?xml version="1.0" standalone="yes"?> يعتبر إرشاداً للمعالجة وليس عنصر". وتوضح تعليمات البرمجة ٨-١ هذا المستند، والآن فقط لتعريف نوع المستند. ويقوم إعلان نوع المستند بالإعلان عن أن العنصر الجذري هو GREETING. ويحتوي إعلان نوع المستند أيضاً على تعريف نوع المستند، الذي يعلن أن عنصر GREETING يحتوي على بيانات أحرف موزعة.

تعليمات البرمجة ٨-١: Hello XML يحتوي على DTD

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE GREETING [
  <!ELEMENT GREETING (#PCDATA)>
]>
<GREETING>
Hello XML!
</GREETING>
```

ويتمثل الفارق الوحيد بين تعليمات البرمجة ٣-٢ و٨-١ في الأسطر الثلاثة الجديدة المضافة إلى تعليمات البرمجة ٨-١ وهي:

```
<!DOCTYPE GREETING [
  <!ELEMENT GREETING (#PCDATA)>
]>
```

وهذه الأسطر هي إعلان نوع المستند لتعليمات البرمجة ٨-١. ويأتي لإعلان نوع المستند بين إعلان XML والمستند نفسه. ويطلق على كل من إعلان نوع المستند وإعلان XML معاً prolog أو التسجيل الأولي للمستند وفي المثال القصير السابق كان إعلان XML هو <?xml version="1.0" standalone="yes"?> is the XML وإعلان نوع المستند هو <!DOCTYPE GREETING [<!ELEMENT GREETING (#PCDATA)>]> وكان <!ELEMENT GREETING (#PCDATA)> هو تعريف نوع المستند وأما <GREETING> Hello XML! </GREETING> فكان هو العنصر الجذري للمستند.

ويبدأ إعلان نوع المستند بعلامة <!DOCTYPE وينتهي بعلامة]>. ومن المتعارف عليه في هذا الأمر وضع البداية والنهاية على سطرين منفصلين، إلا أن فواصل الأسطر والمساحات الفارغة الإضافية ليست هامة. ومن الممكن كتابة إعلان نوع المستند نفسه على سطر منفرد كما يلي:

```
<!DOCTYPE GREETING [ <!ELEMENT GREETING (#PCDATA)> ]>
```

ويتبع اسم العنصر الجذري، GREETING، في هذا المثال <!DOCTYPE. وليس هذا مجرد اسم بل هو شيء مطلوب ولابد لأي مستند صحيح يحمل تعريف المستند هذا أن يكون له العنصر الجذري GREETING. وفيما بين] و[يكمن تعريف نوع المستند.

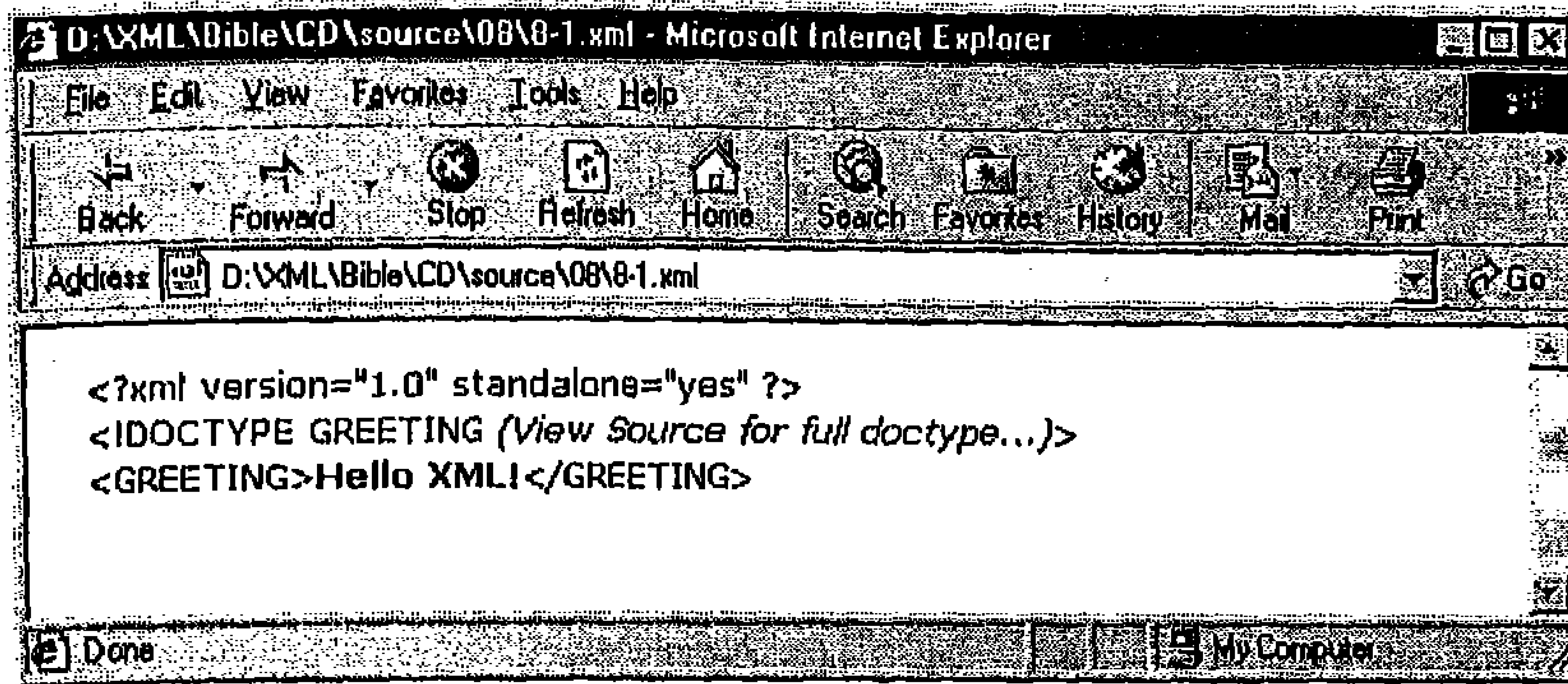
ويتكون DTD من سلسلة من إعلانات الترميز التي تقوم بإعلان عناصر وكينونات وسمات معينة. ويقوم أحد هذه الإعلانات بإعلان العنصر الجذري. ويتواجد DTD الكامل في هذا السطر التالي من تعليمات البرمجة ٨-١:

```
<!ELEMENT GREETING (#PCDATA)>
```

وبصفة عامة تكون DTDs أطول وأكثر تعقيداً بالطبع.

ويعتبر السطر الفردي <!ELEMENT GREETING (#PCDATA)> "تحسن حالة الأحرف مثل معظم الأشياء الموجودة في XML" هو إعلان نوع العنصر element type declaration. وفي هذه الحالة يكون اسم العنصر المعلن هو GREETING. وهو يعتبر العنصر الوحيد. وقد يحتوي هذا العنصر على بيانات أحرف موزعة "#PCDATA". وتعتبر بيانات الأحرف الموزعة هي أي نص ليس نص ترميز. ويشتمل أيضاً على مراجع كينونات مثل &amp ويتم استبدال ذلك بنص عندما يكون النص موزعاً.

ويمكنك تحميل هذا المستند داخل أحد مستعرضات XML بالطريقة المعتادة. ويوضح الشكل ٨-١ تعليمات البرمجة ٨-١ في Internet Explorer 5.0. وستكون النتائج تقريباً كما تتوقعها، وهي عرض مفصل قابل للطي لمصدر المستند. ويشير Internet Explorer إلى أن إعلان نوع المستند يتم تقديمه بإضافة سطر باللون الأزرق.



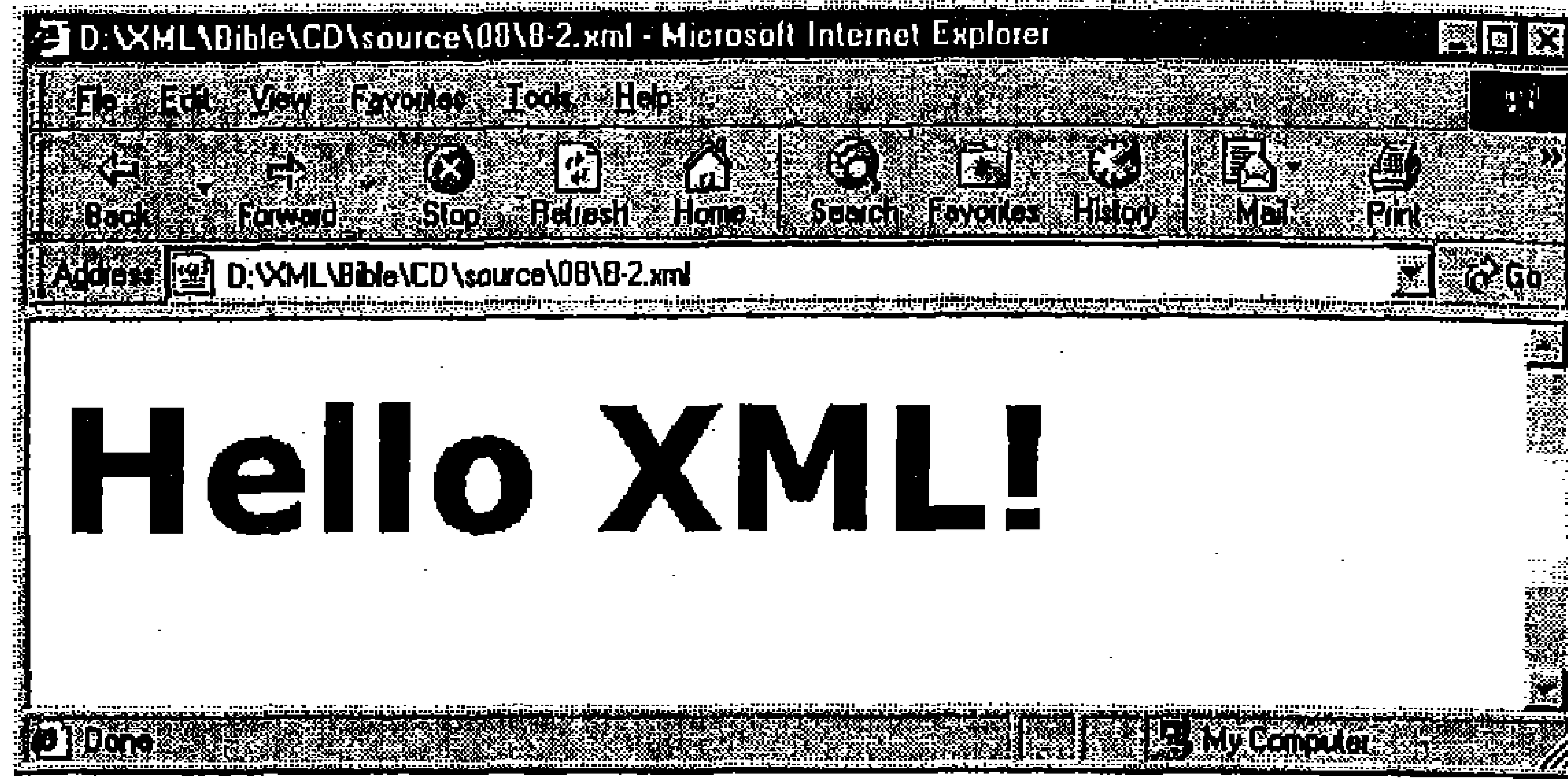
الشكل ٨-١ Hello XML مع DTD عند عرضه في Internet Explorer 5.0.

وبالطبع يمكن ضم المستند بواسطة ورقة نمط تماماً كما في تعليمات البرمجة ٣-٦ في الفصل ٣ وفي الواقع يمكنك استخدام نفس ورقة النمط وعليك فقط إضافة إرشادات المعالجة المعتادة <?xml-stylesheet?> إلى التسجيل الأولى كما هو موضح في تعليمات البرمجة ٨-٢.

تعليمات البرمجة ٨-٢: Hello XML يحتوي على DTD وورقة النمط

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="greeting.css"?>
<!DOCTYPE GREETING [
  <!ELEMENT GREETING (#PCDATA)>
]>
<GREETING>
Hello XML!
</GREETING>
```

ويوضح الشكل ٨-٢ صفحة الويب الناتجة وهي نفس الموضحة في الشكل ٣-٣ في الفصل ٣ تماماً، بدون DTD. وبصفة عامة لا يراعي التنسيق DTD.



الشكل ٨-٢ Hello XML مع DTD وورقة النمط عند عرضه فـي Internet Explorer 5.0.

التحقق من صحة تعريف نوع المستند

من الضروري أن يفي المستند بالقيود التي يتم تعيينها بواسطة DTD وكذلـك لابـد أن يكـون العنصر الجذري له هو الذي يتم تعيينه في إعلان نوع المستند. وكما هـو موضـح بتعليمـات البرمجة ٨-١ لإعلان نوع المستند وDTD، فإنه من المفترض أن يبدو المستند الصحيح كما يلي:

```
<GREETING>
  various random text but no markup
</GREETING>
```

ولا يبدو المستند الصحيح كما يلي:

```
<GREETING>
  <sometag>various random text</sometag>
  <someEmptyTag/>
</GREETING>
```

ولا كما يلي:

```
<GREETING>
  <GREETING>various random text</GREETING>
</GREETING>
```

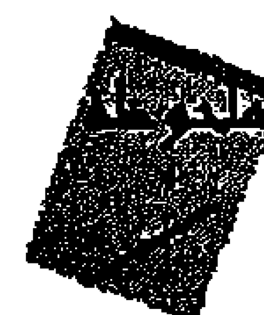
ولا ينبغي أن يتكون المستند مما هو أكثر أو أقل من بيانات أحرف موزعة بين علامة الفتـح <GREETING> وعلامة إغلاق </GREETING>. وعلى العكس من المستند صحيح التكوين، فإن المستند الصحيح لا يسمح بوجود العلامات العشوائية. ولا بد أن يتم الإعلان عن أية علامات

مستخدمة، وذلك في DTD الخاص بالمستند وكذلك ينبغي استخدام هذه العلامات فقط بالطريقة المتاحة بواسطة DTD. وفي تعليمات البرمجة ٨-١، يمكن استخدام علامة <GREETING> فقط لبدء العنصر الجذري ومن المحتمل ألا يتم تضمينه.

وبفرض أننا قمنا بإجراء تغيير بسيط إلى تعليمات البرمجة ٨-٢ باستبدال علامتي <GREETING> و</GREETING> بعلامتي <foo> و</foo>، كما هو موضح بتعليمات البرمجة ٨-٣. وتعليمات البرمجة ٨-٣ هي تعليمات غير معالجة. وهي تعتبر مستند صحيح التكوين إلا أنه لا يفي بالقيود التي يحددها إعلان نوع المستند وDTD التي يحتويها المستند.

تعليمات البرمجة ٨-٣: Hello XML غير صالح لا يطابق قواعد DTD

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="greeting.css"?>
<!DOCTYPE GREETING [
  <!ELEMENT GREETING (#PCDATA)>
]>
<foo>
Hello XML!
</foo>
```



ليس من الضروري أن تكون كل المستندات سليمة. ولا تقوم كل الموزعات بالتحقق من صحة المستندات. وفي الواقع، لا تتحقق معظم المستعرضات التي تشتمل على IE5 وmozilla من صحة المستندات.

ويقوم موزع التحقق من الصحة بقراءة DTD والتحقق مما إذا كان المستند يلتزم بالقواعد المحددة بواسطة DTD وإذا كان المستند يلتزم بها، فإن الموزع يقوم بتمرير البيانات مع تطبيق XML "مثل مستعرض الويب أو قاعدة البيانات". وإذا وجد الموزع أية أخطاء، فإنه يقوم بتقرير الخطأ. وإذا كنت تكتب XML يدوياً، فستحتاج إلى التحقق من صحة مستنداتك قبل توزيعها، حتى تطمئن تماماً إلى أن القراء لن تصادفهم أية أخطاء.

وهناك ما يزيد على عشر موزعات للتحقق من الصحة متاحة على الويب. ومعظمها يكون مجانياً. وكذلك يعتبر معظمها مكتبات يلجأ إليها المبرمجون ليقوموا بالدمج داخل منتجاتهم الخاصة والتي تم إنجازها، ويكون لديهم واجهات المستخدمين الدنيا "إن وجدت". وتشتمل

الموزعات فـــي هــذه الفئــة علـــى IBM's alphaWorks' XML for Java و Microsoft و dataChannel's XJParser و Silfide's SXP.

XML for Java: http://www.alphaworks.ibm.com/tech/xml
XJParser: http://www.datachannel.com/xml_resources/
SXP: http://www.loria.fr/projets/XSilfide/EN/sxp/

وتتضمن بعض المكتبات أيضاً على موزعات مستقلة بذاتها والتي تعمل مــن خــلال ســطر الأوامر. وهي تعتبر برامج تقوم بقراءة ملف XML وتقوم بتقرير أية أخطاء موجودة، مع عـــدم عرضها. فعلى سبيل المثال، يعتبر XJParse أحد برامج Java المضمنة عــن طريــق IBM's XML for Java فئة 1.1.16 في مجموعة samples.XJParse. ولتشغيل هذا البرنامج، يجــب عليك إضافة ملفات XML for Java إلى مسار فئة Java الخاص بك. وعندئذ يمكنك التحقق مــن صحة أحد الملفات بفتح DOS Window أو مطالبة هيكلية وتحويــل الاســم المحلــي أو URL المنعزل للملف الذي تريد التحقق من صحته إلى برنامج XJParse، كما يلي:

```
C:\xml4j>java samples.XJParse.XJParse -d D:\XML\08\invalid.xml
```

في أثناء كتابة هذه السطور قام IBM's alphaWorks بإصدار 2.0.6 من XML for Java. وفي هذا الإصـــدار تقــوم باســتدعاء فقــط XJParse بــدلاً مــن samples.XJParse. إلا أن إصدار 1.1.16 يقدم ميزات أكثر لقواعد التحقــق من الصحة المستقلة.

ويمكنك استخدام URL بدلاً من اسم الملف، كما هو موضح أسفل:

```
C:\xml4j>java samples.XJParse.XJParse -d
http://metalab.unc.edu/books/bible/examples/08/invalid.xml
```

وفي أي من الحالات، يقوم XJParse بالرد بواسطة قائمة بالأخطـــاء الموجــودة، متبوعــة بنموذج متفرع للمستند فعلى سبيل المثال:

```
D:\XML\07\invalid.xml: 6, 4: Document root element, "foo", must
match DOCTYPE root, "GREETING".
D:\XML\07\invalid.xml: 8, 6: Element "<foo>" is not valid in
this context.
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css"  href="greeting.css"?>
<!DOCTYPE GREETING [
```

```
<!ELEMENT GREETING (#PCDATA)>
]>
<foo>
Hello XML!
</foo>
```

ولا يعتبر هذا بصفة خاصة ناتج فعال إلا أن الغرض من موزعات التحقق من الصحة مثل XJParse ليس عرض ملفات XML. فوظيفة الموزع هي تقسيم المستند إلى داخل بنية متفرعة ونقل عقد الشجرة إلى البرنامج الذي يقوم بعرض البيانات. وقد يكون هذا البرنامج مستعرض ويب مثل Netscape Navigator أو Internet Explorer. وقد يكون حتى برنامج مخصص قمت بكتابته بنفسك. ويستخدم XJParse أو أي سطر أوامر آخر أو موزع التحقق من الصحة للتحقق من كتابتك مستند XML جيد يمكن للبرامج الأخرى التعامل معه. وفي الأساس، يعتبر ذلك شكل من أشكال بروفات الطباعة أو تأمين الجودة وليس ناتج متكامل.

وحيث أن XML for Java ومعظم موزعات التحقق من الصحة الأخرى تتم كتابتها في Java، فإنها تجلب كل عيوب برامج النظام الترافقي Java. وفي البداية، وقبل أن تتمكن من تشغيل الموزع، لابد أن يكون مثبت لديك إما "JDK" Java Development Kit أو Java Runtime Environment. وثانياً، ستحتاج إلى إضافة ملفات XML for Java إلى مسار الفئة الخاص بك. وليس من السهل القيام بكل من هاتين المهمتين. ولم يتم تصميم أي من الأداتين بعين المستخدم غير المبرمج، بل هما مصممتان فقط لاستخدام عديم الجدوى.

وإذا كنت تكتب مستندات إلى مستعرضات ويب فستكون أسهل الطرق للتحقق من صحتها هي تحميلها داخل المستعرض ورؤية الأخطاء التي يقررها. ولا تتحقق كل مستعرضات الويب من صحة المستندات. فبعضها قد تقبل ببساطة المستندات صحيحة التكوين دون الاهتمام بالتحقق من صحتها ويتحقق Internet Explorer 5.0 من صحة المستندات.

وهناك بديل إذا كانت المستندات موجودة على خادم الويب، وليست خاصة بشكل واضح، وهو المحققات الموجودة على الويب للتحقق من الصحة. وتتطلب هذه الموزعات فقط أن يقوم المستخدم بإدخال URL الخاص بالمستند. ولهذه الموزعات ميزة مميزة جداً وهي عدم مطالبة المستخدم بالخوض في برنامج وقت تشغيل Java ومسارات الفئات ومتغيرات البيئة.

ويوضح الشكل ٨-٣ XML well-formedness checker and validator الموجود على موزع RXP والمستضاف على الويب. وسوف تجده في http://www.cogsci.ed.ac.uk/%7Erichard/xml-check.html. وأما الشكل ٨-٤ فيوضح الأخطاء المعروضة كنتائج لاستخدام هذا البرنامج للتحقق من صحة تعليمات البرمجة ٨-٣.

ويقدم Brown University's Scholarly Technology Group محققاً من الصحـــة في http:// www.stg.brown.edu ومن الجدير بالذكر أنه يسمح للمستخدم بتحميل ملفــات مــن الكمبيوتر الشخصي بدلاً من وضعها على خادم ويب محلي وهو موضح بالشــكل ٨-٥. وأمــا الشكل ٨-٦ فيوضح نتائج استخدام هذا البرنامج للتحقق من صحة تعليمات البرمجة ٨-٣.

XML well-formedness checker and validator - Netscape

http://www.cogsci.ed.ac.uk/%7Erichard/xml-check.html

XML well-formedness checker and validator

Use this form to check an XML document for well-formedness and (optionally) validity. The checker uses the RXP parser. External entity references are included.

If the document is well-formed, the parser outputs the corresponding canonical XML.

Only HTTP URLs are allowed.

http://sunsite.unc.edu/xml/invalid.xml [check it] [clear]

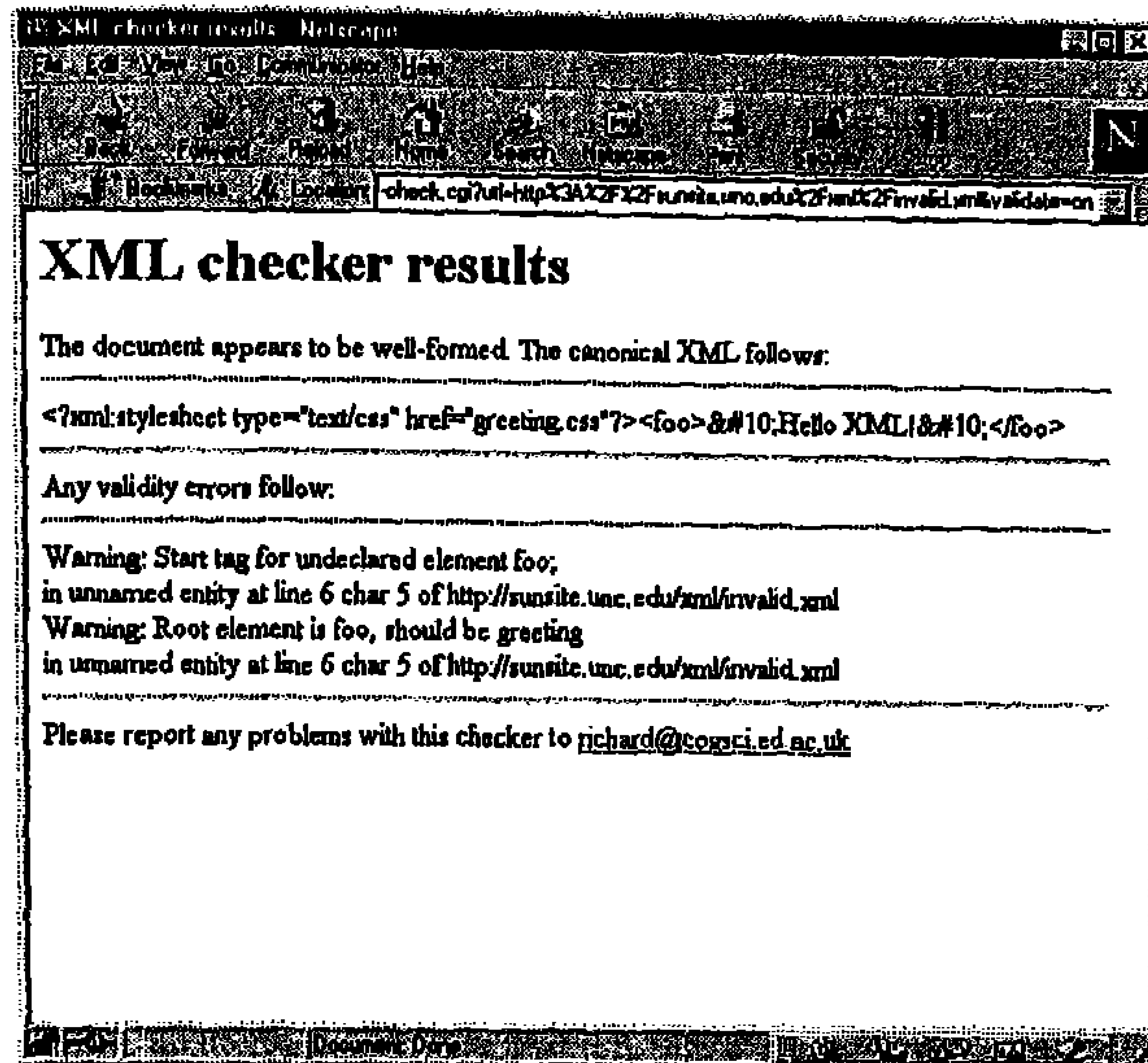
☑ validate?

Some other XML checkers are:

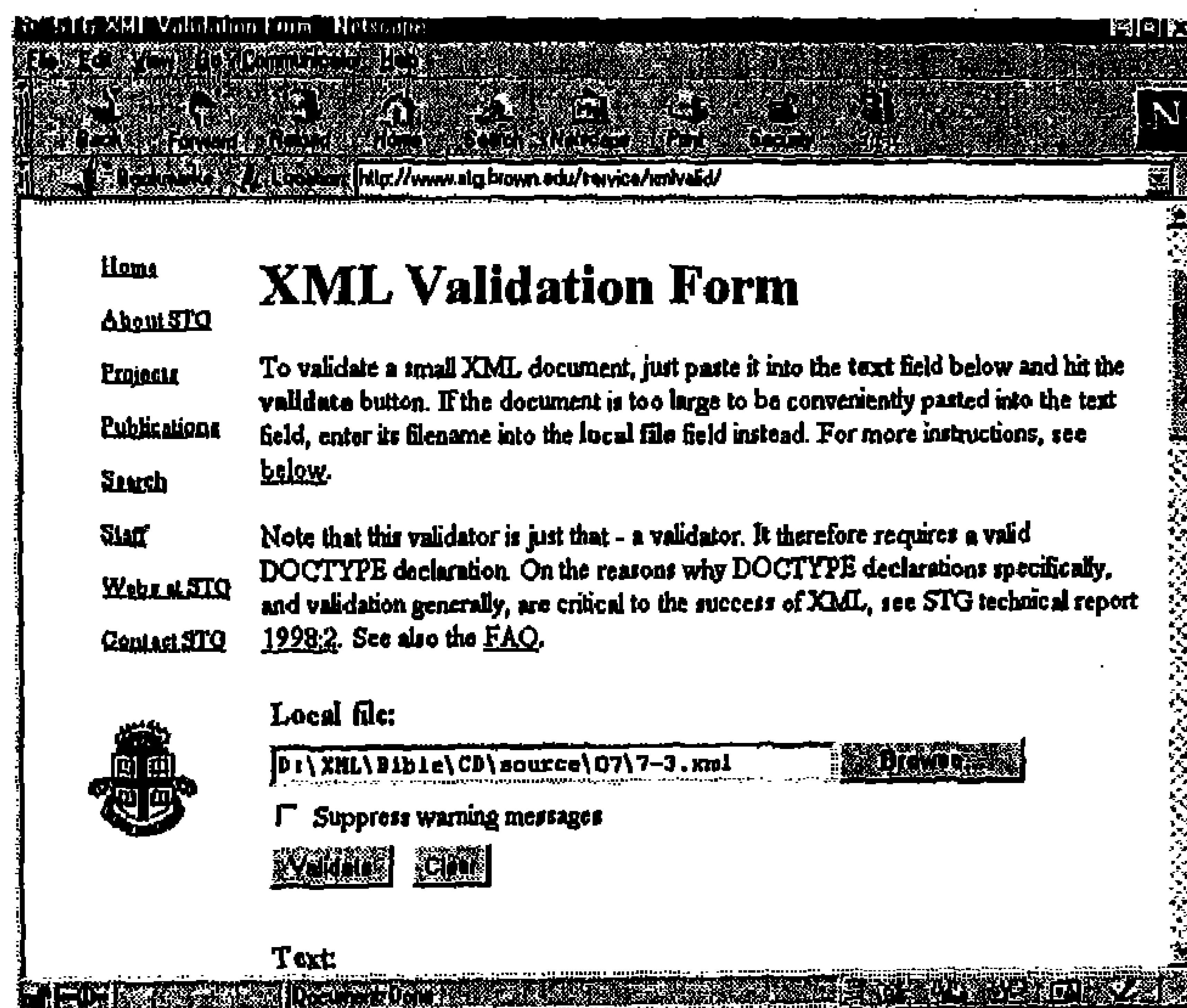
- Frontier 5
- RUWF
- STG's validator
- Koala XML validator

Document: Done

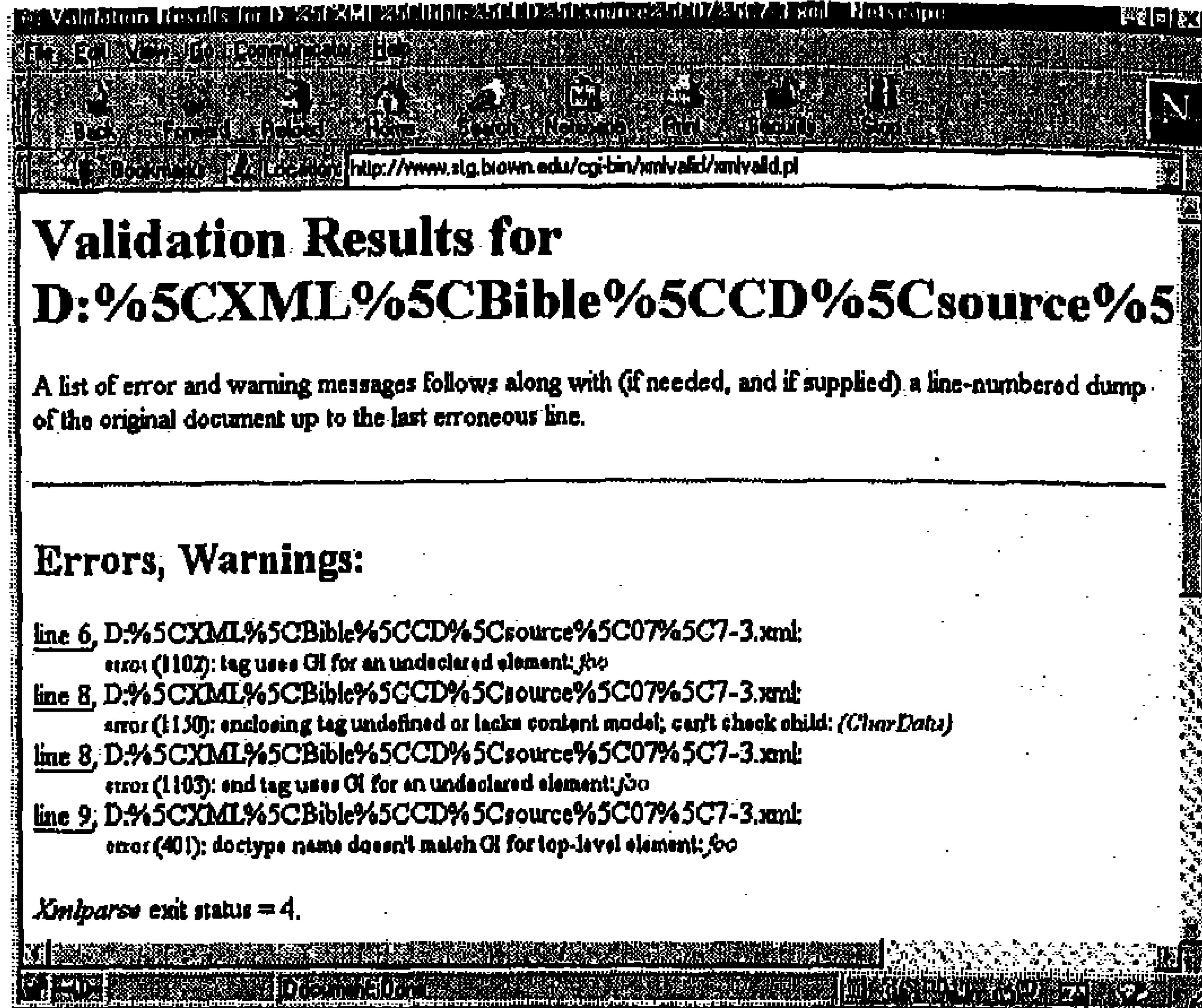
الشكل ٨-٣ Richard Tobin's الموجودة على مـــوزع RXP و XML well-formedness checker and validator المستضافة على الويب.



الشكل ٨-٤ الأخطاء الموجودة في تعليمات البرمجـــة ٨-٣ كما تم التقريـــر عنــها باســتخدام Richard Tobin's XML validator.



الشكل ٨-٥ Brown University's Scholarly Technology Group's Web-hosted XML validator والمستضاف عبر خدمات الويب.



الشكل ٨-٦ الأخطاء الموجودة في تعليمات البرمجة ٨-٣ كما في تقرير Brown University's Scholarly Technology Group's Web-hosted XML validator.

سرد العناصر

تعتبر الخطوة الأولى لإنشاء DTD مناسب لمستند معين هي فهم بنية المعلومات التي يتم وضعها على هيئة تعليمات برمجية باستخدام العناصر المعرفة في DTD وفي بعض الأحيان تكون المعلومات بنية دقيقة كما في قائمة جهات الاتصال. وفي الأحيان الأخرى تكون ذات شكل حر إلى حد ما كما في القصص القصيرة المصورة أو مقال في مجلة.

وسنتخذ الآن مستند ذو بنية كمثال لنا. وبالتحديد سنعود إلى إحصاءات البيسبول الموضحة في الفصل ٤. وتتيح إضافة DTD إلى هذا المستند لنا تنفيذ القيود التي تم الالتزام بها فيما سبق فقط عند التحويل. فعلى سبيل المثال، يمكننا أن نطلب أن يحتوي SEASON بالتحديد على تابعي LEAGUE، وأن يحتوي كل TEAM على TEAM_CITY وTEAM_NAME وأن يسبقه TEAM_CITY دائماً TEAM_NAME.

وتذكر أن مستند إحصاءات البيسبول المتكاملة يحتوي على العناصر التالية:

SEASON
YEAR
LEAGUE
LEAGUE_NAME
DIVISION
DIVISION_NAME
TEAM
TEAM_CITY
TEAM_NAME
PLAYER
SURNAME
GIVEN_NAME
POSITION
GAMES
GAMES_STARTED
AT_BATS
RUNS
HITS
DOUBLES
TRIPLES
HOME_RUNS
WINS
LOSSES
SAVES
RBI
STEALS
CAUGHT_STEALING
SACRIFICE_HITS
SACRIFICE_FLIES
ERRORS
WALKS
STRUCK_OUT
HIT_BY_PITCH
COMPLETE_GAMES
SHUT_OUTS
ERA
INNINGS
HOME_RUNS
RUNS
EARNED_RUNS
HIT_BATTER
WILD_PITCHES
BALK
WALKED_BATTER
STRUCK_OUT_BATTER
COMPLETE_GAMES
SHUT_OUTS

ويحتاج DTD الذي تكتبه إلى إعلانات عناصر لكل منها ويقوم كل إعلان عنصر بسرد اسم العنصر والتوابع التي قد يحتوي عليها العنصر. فعلى سبيل المثال، يتطلب DTD أن يحتوي LEAGUE بالتحديد على ثلاثة توابع DIVISION. وقد يتطلب أيضاً أن يكون عنصر SURNAME داخل عنصر PLAYER وليس خارجه أبداً. وقد يصر على أن يكون لتابع DIVISION عدد غير محدد من عناصر TEAM أو ليس أقل أبداً من عنصر واحد.

وقد يتطلب DTD أن يكون لعنصر PLAYER بالتحديد عنصر واحد من كل عناصر GIVEN_NAME وSURNAME وPOSITION وGAMES، إلا أنه يجعل ما إذا كان عنصر PLAYER يحتوي على RBI أم ERA أمراً خيارياً. وكذلك من الممكن أن تتطلب أن يتم استخدام

عناصر GIVEN_NAME وSURNAME وPOSITION وGAMES بترتيب معين. وأيضـــاً قــد يتطلب DTD أن تتواجد العناصر في سياق معين. فعلى سبيل المثـــال، قــد تســتخدم عنــاصر GIVEN_NAME وSURNAME وPOSITION وGAMES فقط داخل عنصر PLAYER.

ويكون البدء أسهل غالباً إذا كان فكرة لنموذج مستند متكامل وصحيح التكوين يســتخدم كــل العناصر التي تريدها في DTD. وتخدم الأمثلة الموجودة في الفصل ٤ هذا الغـــرض. وتعتــبر تعليمات البرمجة ٨-٤ إصدار منسق من تعليمات البرمجة ٤-١ الموجودة في الفصل ٤ وبالرغم من احتوائها على لاعبين فقط، فهي توضح كل العناصر الهامة.

تعليمات البرمجة ٨-٤: مستند XML صحيح التكوين سيتم كتابة DTD له

```
<?xml version="1.0" standalone="yes"?>
<SEASON>
 <YEAR>1998</YEAR>
 <LEAGUE>
  <LEAGUE_NAME>National</LEAGUE_NAME>
  <DIVISION>
   <DIVISION_NAME>East</DIVISION_NAME>
    <TEAM>
     <TEAM_CITY>Florida</TEAM_CITY>
     <TEAM_NAME>Marlins</TEAM_NAME>
     <PLAYER>
      <SURNAME>Ludwick</SURNAME>
      <GIVEN_NAME>Eric</GIVEN_NAME>
      <POSITION>Starting Pitcher</POSITION>
      <WINS>1</WINS>
      <LOSSES>4</LOSSES>
      <SAVES>0</SAVES>
      <GAMES>13</GAMES>
      <GAMES_STARTED>6</GAMES_STARTED>
      <COMPLETE_GAMES>0</COMPLETE_GAMES>
      <SHUT_OUTS>0</SHUT_OUTS>
      <ERA>7.44</ERA>
      <INNINGS>32.2</INNINGS>
```

```
    <HOME_RUNS>46</HOME_RUNS>
    <RUNS>7</RUNS>
    <EARNED_RUNS>31</EARNED_RUNS>
    <HIT_BATTER>27</HIT_BATTER>
    <WILD_PITCHES>0</WILD_PITCHES>
    <BALK>2</BALK>
    <WALKED_BATTER>0</WALKED_BATTER>
    <STRUCK_OUT_BATTER>17</STRUCK_OUT_BATTER>
  </PLAYER>
  <PLAYER>
    <SURNAME>Daubach</SURNAME>
    <GIVEN_NAME>Brian</GIVEN_NAME>
    <POSITION>First Base</POSITION>
    <GAMES>10</GAMES>
    <GAMES_STARTED>3</GAMES_STARTED>
    <AT_BATS>15</AT_BATS>
    <RUNS>0</RUNS>
    <HITS>3</HITS>
    <DOUBLES>1</DOUBLES>
    <TRIPLES>0</TRIPLES>
    <HOME_RUNS>0</HOME_RUNS>
    <RBI>3</RBI>
    <STEALS>0</STEALS>
    <CAUGHT_STEALING>0</CAUGHT_STEALING>
    <SACRIFICE_HITS>0</SACRIFICE_HITS>
    <SACRIFICE_FLIES>0</SACRIFICE_FLIES>
    <ERRORS>0</ERRORS>
    <WALKS>1</WALKS>
    <STRUCK_OUT>5</STRUCK_OUT>
    <HIT_BY_PITCH>1</HIT_BY_PITCH>
  </PLAYER>
</TEAM>
<TEAM>
  <TEAM_CITY>Montreal</TEAM_CITY>
```

```
      <TEAM_NAME>Expos</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>New York</TEAM_CITY>
      <TEAM_NAME>Mets</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Philadelphia</TEAM_CITY>
    <TEAM_NAME>Phillies</TEAM_NAME>
    </TEAM>
</DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Chicago</TEAM_CITY>
       <TEAM_NAME>Cubs</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Arizona</TEAM_CITY>
       <TEAM_NAME>Diamondbacks</TEAM_NAME>
     </TEAM>
  </DIVISION>
 </LEAGUE>
 <LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Baltimore</TEAM_CITY>
       <TEAM_NAME>Orioles</TEAM_NAME>
     </TEAM>
```

```
      </DIVISION>
      <DIVISION>
        <DIVISION_NAME>Central</DIVISION_NAME>
         <TEAM>
           <TEAM_CITY>Chicago</TEAM_CITY>
           <TEAM_NAME>White Sox</TEAM_NAME>
         </TEAM>
      </DIVISION>
      <DIVISION>
        <DIVISION_NAME>West</DIVISION_NAME>
         <TEAM>
           <TEAM_CITY>Anaheim</TEAM_CITY>
           <TEAM_NAME>Angels</TEAM_NAME>
         </TEAM>
       </DIVISION>
    </LEAGUE>
  </SEASON>
```

ويسرد الجدول ٨-١ عناصر مختلفة في هذا المثال ولابد أيضاً من الالتزام بالشروط. ويحتوي كل عنصر على قائمة بالعناصر التي يجب أن يشتمل عليها، وكذلك العنصر الذي ينبغي احتواؤه به. وقد يشتمل العنصر، في بعض الحالات، على أكثر من عنصر واحد تابع من نفس النوع. ويحتوي SEASON على YEAR واحد وعنصري LEAGUE ويحتوي DIVISION بصفة عامة على أكثر من TEAM واحد. وبصورة أقل وضوحاً يقوم بعض المهاجمين بالتبديل بين ضارب كرة معين وأقصى الملعب من لعبة إلى لعبة. لذلك قد يحتوي عنصر PLAYER واحد على أكثر من POSITION واحد. وفي الجدول الموضح تم الإشارة إلى المطالبة بعدد معين من التوابع عن طريق بدء العنصر بعدد "على سبيل المثال LEAGUE 2" ويشار إلى إمكانية تعدد التوابع بإضافة "s" إلى نهاية اسم العنصر مثل PLAYER(s).

وتلتزم تعليمات البرمجة ٨-٤ بهذه الشروط. وكان من الممكن أن تكون أقصر من ذلك إذا تم تجاهل عنصري PLAYER وبعض عناصر TEAM. وكان من الممكن أن تكون أطول إذا تم تضمين عناصر PLAYER أخرى عديدة. إلا أنه من المطلوب أن تكون كل العناصر الأخرى في المواضع التي تظهر بها.

للعناصر نوعان أساسيان في XML. عناصر بسيطة تحتوي على النص، والمعروفة أيضاً باسم بيانات الأحرف الموزعة PCDATA# أو PCDATA بهذا السياق. وعناصر مركبة تحتوي على العناصر الأخرى أو نادراً ما تحتوي على النص والعناصر الأخرى ولا يوجد أعداد صحيحة أو فواصل عائمة "منتقلة" أو تاريخ أو البيانات الأخرى في XML. لذلك فلن يمكنك استخدام DTD في توضيح أن عدد التحركات لابد وأن يكون عدد صحيح غير متنقل، أو أن ERA لابد أن يكون عدد فاصلة متنقلة بين 0.0 و1.0، حتى بالرغم من أن القيام بذلك سيكون مفيداً في مثل هذه الأمثلة. وهناك جهود مبذولة منذ فترة لتعريف مخططات تستخدم عبارة XML لوصف معلومات قد تتم كتابتها على هيئة تعليمات برمجية في DTD بصورة عادية وأيضاً معلومات نوع البيانات. وبمنتصف عام ١٩٩٩ تكون هذه الأكثر نظرياً باستخدام بضعة تطبيقات عملية.

والآن وبعد تعريفك للمعلومات التي تقوم بتخزينها، وكذلك العلاقات المطلوبة والخيارية بين هذه العناصر، فأنت على استعداد لإنشاء DTD للمستند الذي يقوم بدقة بتلخيص هذه العلاقات.

وغالباً ما يكون من الممكن والمناسب القص واللصق من أحد DTDs إلى آخر. ويمكن إعادة استخدام العديد من العناصر مرة أخرى في سياقات أخرى. فعلى سبيل المثال، يعمل وصف TEAM بصورة متكافئة وجيدة بالنسبة لكرة القدم والهوكي وأغلب فرق الرياضة الأخرى.

ويمكنك تضمين أحد DTDs داخل آخر حتى يرسم المستند علامات من كل منهما. وقد تستخدم، على سبيل المثال، DTD يصف إحصاءات لاعبين معينين بالتفصيل ثم تضمين DTD هذا داخل DTD الشامل لفرق الرياضة. وللتغيير من البيسبول إلى كرة القدم، قم ببساطة بتصنيف DTD للاعب البيسبول إلى DTD للاعب كرة قدم.

للقيام بذلك، ستجد الملف الذي يحتوي على DTD معرفاً كمكون خارجي. ويوضح الفصل ٩ "الكينونات" مراجع مكونات المعاملات الخارجية.

الجدول ٨ - ١

العناصر الموجودة في إحصاءات البيسبول

العنصر	العناصر التي يجب أن يحتوي عليها	العناصر التي قد يحتوي عليها	العناصر التي يجب أن تحتويه "إن وجدت"
SEASON	YEAR,	2 LEAGUE	
YEAR	نص		SEASON
LEAGUE	LEAGUE_NAME 3 DIVISION		SEASON
LEAGUE_NAME	نص		LEAGUE
DIVISION	DIVISION_NA TEAM ME	TEAM(s)	LEAGUE
DIVISION_NAME	نص		DIVISION
TEAM	TEAM_CITY, TEAM_NAME	PLAYER(s)	DIVISION
TEAM_CITY	نص		TEAM
TEAM_NAME	نص		TEAM



الجدول ٨-١

العناصر الموجودة في إحصاءات البيسبول

العنصر	العناصر التي يجب أن يحتوي عليها	العناصر التي قد يحتوي عليها	العناصر التي يجب أن تحتويه "إن وجدت"
PLAYER	SURNAME, GIVEN_NAME, POSITION GAMES	GAMES_STARTED, AT_BATS, RUNS, HITS, DOUBLES, TRIPLES, HOME_RUNS, RBI, STEALS, CAUGHT_STEALING, SACRIFICE_HITS, SACRIFICE_FLIES, ERRORS, WALKS, STRUCK_OUT, HIT_BY_PITCH, COMPLETE_GAMES, SHUT_OUTS, ERA, INNINGS, HIT_BATTER, WILD_PITCHES, BALK, WALKED_BATTER, STRUCK_OUT_BATTER	TEAM

SURNAME	نص	PLAYER
PLAYER	نص	GIVEN_NAME

الجدول ٨-١
العناصر الموجودة في إحصاءات البيسبول

العنصر	العناصر التي يجب أن يحتوي عليها	العناصر التي قد يحتوي عليها	العناصر التي يجب أن تحتويه "إن وجدت"
PLAYER	نص		POSITION
GAMES	نص		PLAYER
GAMES STARTED	نص		PLAYER
AT_BATS	نص		PLAYER
RUNS	نص		PLAYER
HITS	نص		PLAYER
DOUBLES	نص		PLAYER
TRIPLES	نص		PLAYER
HOME_RUNS	نص		PLAYER
RBI	نص		PLAYER
STEALS	نص		PLAYER
CAUGHT_ STEALING	نص		PLAYER
SACRIFICE_ HITS	نص		PLAYER
SACRIFICE_FLIES	نص		PLAYER
ERRORS	نص		PLAYER
WALKS	نص		PLAYER
STRUCK_OUT	نص		PLAYER
HIT_BY_ PITCH	نص		PLAYER

الجدول ٨-١
العناصر الموجودة في إحصاءات البيسبول

العنصر	العناصر التي يجب أن يحتوي عليها	العناصر التي قد يحتوي عليها	العناصر التي يجب أن تحتويه "إن وجدت"
COMPLETE_GAMES	نص		PLAYER
SHUT_OUTS	نص		PLAYER
ERA	نص		PLAYER
INNINGS	نص		PLAYER
HOME_RUNS_AGAINST	نص		PLAYER
INNINGS	نص		PLAYER
HOME_RUNS_AGAINST	نص		PLAYER
RUNS_AGAINST	نص		PLAYER
HIT_BATTER	نص		PLAYER
WILD_PITCHES	نص		PLAYER
BALK	نص		PLAYER
WALKED_BATTER	نص		PLAYER
STRUCK_OUT_BATTER	نص		PLAYER

إعلانات العناصر

لابد أن يتم إعلان كل علامة مستخدمة في مستند XML صحيح باستخدام إعلان عنصر في DTD. ويحدد إعلان العنصر اسم العنصر وما يمكن أن يحتويه. وفي بعض الأحيان، يطلق على قائمة المحتويات "مواصفة المحتوى". وتستخدم مواصفة المحتوى تدقيق نحوي بسيط لتحديد ما

هو مسموح وما هو غير مسموح به في المستند بدقة وقد يبدو ذلك شيئاً شـــديد التعقيـد، إلا أن الأمر هو أنك تقوم بإضافة علامة تنقيط، مثل، "*" أو "؟" أو "+" إلى اسم عنصر للإشـارة إلـى احتمال تواجده أكثر من مرة أو احتمال عدم تواجده أو وجوب تواجده مرة على الأقل.

وتعتبر DTD إحدى وسائل الوقاية. فكل ما هو غير متاح على نحو واضح، يكون محظـوراً إلا أن بناء جملة DTD يتيح لك تحديد العلاقات التي يصعب تحديدها في الجمل، وذلك بطريقـة محكمة. فعلى سبيل المثال، تسهل عليك DTDs توضيح أن GIVEN_NAME ينبغي أن يأتي قبل SURNAME، والذي ينبغي أن يأتي قبل POSITION، والذي بدوره يأتي قبل GAMES، والـذي من المفترض أن يأتي قبل GAMES_STARTED والذي يجب أن يأتي قبل AT_BATS، والـذي يجب وجوده قبل RUNS الذي يأتي قبل HITS وأن كل ذلك قد يظهر فقط داخل PLAYER.

ومن الأسهل إنشاء DTDs عن طريق العمل الداخلي. ويتيح لك ذلك إنشاء مستند نموذجي في نفس الوقت الذي تقوم فيه بإنشاء DTD للتحقق من أن DTD نفسه صحيح ويصف التنسيق الـذي تريده بالفعل.

كلمة ANY الأساسية

أول ما ينبغي القيام به هو تعريف العنصر الجذري. وفي مثال البيسبول، كـان SEASON هـو العنصر الجذري ويحدد إعلان DOCTYPE! ما يلي:

```
<!DOCTYPE SEASON [

]>
```

إلا أن ذلك يوضح فقط أن العلامة الجذرية هي SEASON. ولكنه لا يوضح أي شيء عما قد يحتويه عنصر SEASON أو قد لا يحتويه، الأمر الذي يفرض عليك إعلان عنصـر SEASON في إعلان العنصر. ويتم ذلك باستخدام سطر التعليمات البرمجية التالي:

```
<!ELEMENT SEASON ANY>
```

وتبدأ إعلانات نوع المستند بعبارة <!ELEMENT "تحسن لحالة الأحرف" وتنتهي بحـروف >. وهي تشتمل على اسم العنصر المعلن "وهو SEASON فـي هـذا المثـال" متبوعـاً بمواصفـة المحتوى. وتوضح كلمة ANY الأساسية "مرة أخرى تحسن لحالة الأحـرف" أن كـل العنـاصر المتاحة وبيانات الأحرف الموزعة يمكن أن تكون توابع لعنصر SEASON.

ويعتبر استخدام ANY شائعاً للعناصر الجذرية، وخاصةً المستندات التي ليس لها بنية، إلا أنـه ينبغي تجنبها في أغلب الحالات الأخرى. وبصفة عامة من الأفضل أن تكون شديد الحرص بقـدر الإمكان فيما يختص بمحتوى كل علامة. وعادةً ما يتم إدخال تحسينات على DTDs عن طريـق

تطويرها، ويقصد بذلك تقليل القيود المتعلقة بالوقت عندما تعكس DTDs هذه استخدامها وسـياقها بصورة جيدة من المرة الأولى. لذلك فمن الأفضل البدء بدون قيود وتسهيل الإجراءات المتبعـة فيما بعد.

#PCDATA

بالرغم من احتمال ظهور عناصر داخل المستند، فإنه ينبغي أن يتم الإعلان عَن العناصر التـي تظهر وأول شيء تحتاج إليه هو YEAR. وإعلان العنصر YEAR هو:

```
<!ELEMENT YEAR (#PCDATA)>
```

ويوضح هذا الإعلان أن YEAR قد يحتوي فقط على بيانات الأحرف الموزعـة، أي النـص غير الترميزي. ومن غير المحتمل أن يحتوي على توابع له. لذلك فإن عنصر YEAR هذا يعتـبر صالحاً:

```
<YEAR>1998</YEAR>
```

وعناصر YEAR التالية تعتبر أيضاً صالحة:

```
<YEAR>98</YEAR>
<YEAR>1998 C.E.</YEAR>
<YEAR>
 The year of our lord one thousand,
 nine hundred, &amp; ninety-eight
</YEAR>
```

وحتى YEAR هذا يعتبر صالحاً حيث أن XML لا يحاول تصحيح محتويات PCDATA التـي هي فقط نص لا يحتوي على ترميز.

```
<YEAR>Delicious, delicious, oh how boring</YEAR>
```

إلا أن عنصر YEAR التالي يعتبر غير صالح حيث أنه يحتوي على عناصر تابعة:

```
<YEAR>
 <MONTH>January</MONTH>
 <MONTH>February</MONTH>
 <MONTH>March</MONTH>
 <MONTH>April</MONTH>
 <MONTH>May</MONTH>
 <MONTH>June</MONTH>
 <MONTH>July</MONTH>
```

```
    <MONTH>August</MONTH>
    <MONTH>September</MONTH>
    <MONTH>October</MONTH>
    <MONTH>November</MONTH>
    <MONTH>December</MONTH>
  </YEAR>
```

ويتم تضمين إعلان العنصر SEASON وYEAR في إعلان نوع المستند كما يلي:

```
<!DOCTYPE SEASON [
  <!ELEMENT SEASON ANY>
  <!ELEMENT YEAR (#PCDATA)>
]>
```

وكالمعتاد، فإن التباعد والمسافات البادئة ليست ضرورية، ولا يعتبر الترتيب الذي تظهر عليه إعلانات العناصر ترتيب وثيق الصلة. ويعني إعلان نوع المستند التالي نفس الشيء بالتحديد:

```
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT SEASON ANY>
]>
```

ويوضح كل منهما أن عنصر SEASON قد يحتوي على بيانات بأي ترتيب. وهذا العنصر الآخر المعلن هو فقط YEAR الذي قد يحتوي فقط على بيانات أحرف موزعة. فعلى سبيل المثال، انظر المستند الموضح في تعليمات البرمجة ٨-٥.

تعليمات البرمجة ٨-٥: مستند صالح

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT SEASON ANY>
]>
<SEASON>
  <YEAR>1998</YEAR>
</SEASON>
```

وحيث أن عنصر SEASON قد يحتوي أيضاً على بيانات أحرف موزعة، فبإمكانك إضافة نص آخر خارج YEAR. وهذا ما توضحه تعليمات البرمجة ٨-٦.

تعليمات البرمجة ٨-٦: مستند صالح يحتوي على YEAR ونص عادي

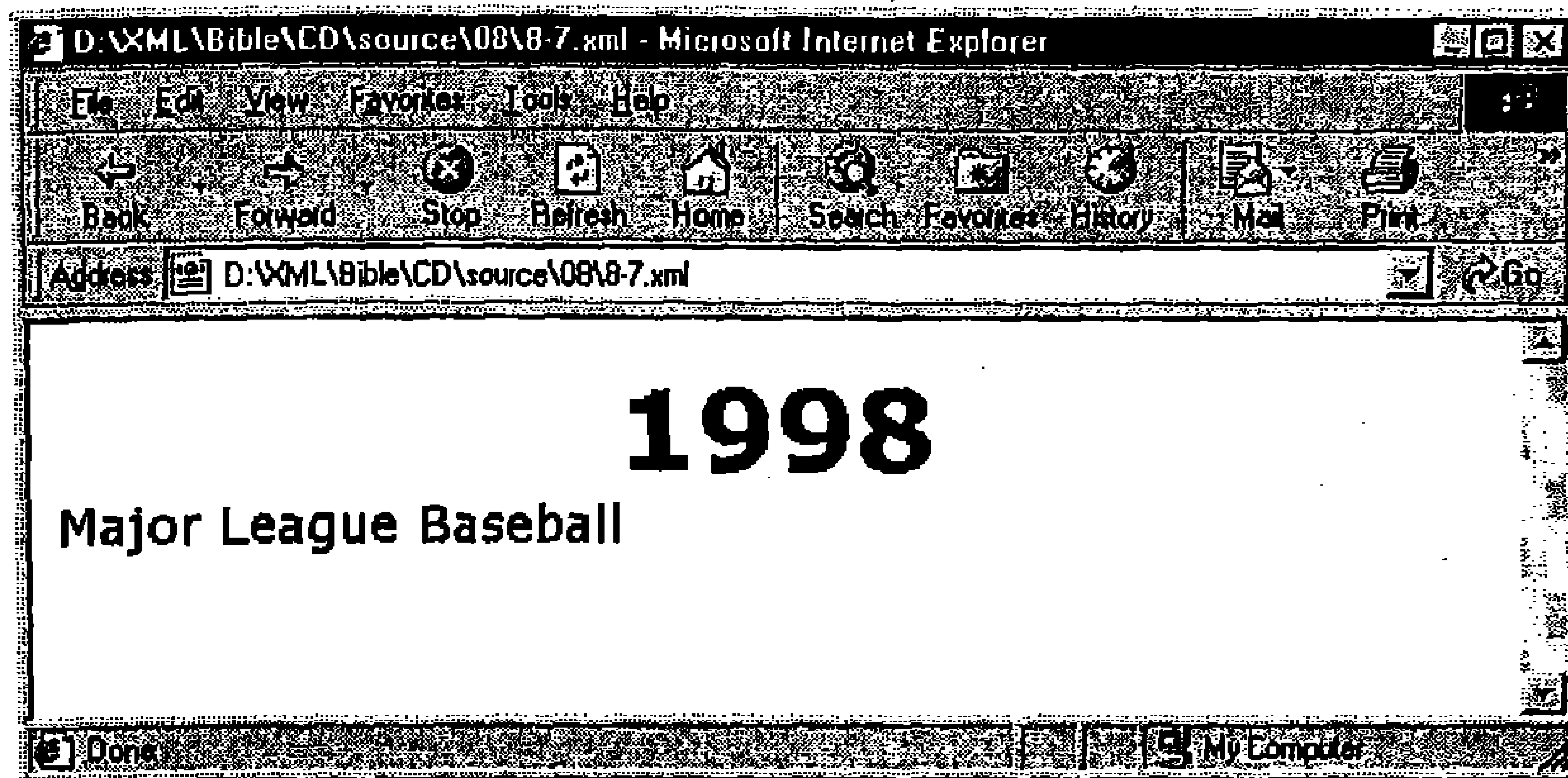
```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT SEASON ANY>
]>
<SEASON>
  <YEAR>1998</YEAR>
  Major League Baseball
</SEASON>
```

وأخيراً سنقوم برفض مثل هذه المستندات. إلا أنه يعتبر الآن سليماً حيث أن SEASON تم إعلانه لقبول محتوى ANY. وغالباً ما يكون من الأسهل البدء باستخدام ANY للعنصر حتى تقوم بتعريف كل توابعه. وعندئذ يمكنك استبداله بالتوابع الفعلية التي تريد استخدامها.

ويمكن إرفاق ورقة نمط بسيطة مثل ورقة نمط baseballstats.css والتي تم تطويرها في الفصل ٤ إلى تعليمات البرمجة ٨-٦، كما هو موضح بتعليمات البرمجة ٨-٧، وتحميلها بمستعرض الويب كما هو موضح بالشكل ٨-٧. وتحتوي ورقة نمط baseballstats.css على قواعد النمط الخاصة بالعناصر غير المقدمة في DTD أو جزء المستند من تعليمات البرمجة ٨-٧، إلا أنها لا تعتبر مشكلة فمستعرضات الويب تتجاهل أية قواعد نمط للعناصر غير الموجودة في المستند.

تعليمات البرمجة ٨-٧: مستند صالح يحتوي على ورقة نمط وYEAR ونص عادي

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="baseballstats.css"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT SEASON ANY>
]>
<SEASON>
  <YEAR>1998</YEAR>
  Major League Baseball
</SEASON>
```



الشكل ٨-٧ مستند صالح يحتوي على ورقة نمـــط وعنصـــر YEAR ونص عادي. يتم عرضه في Internet Explorer 5.0.

قوائم التوابع

حيث أنه تم الإعلان عن عنصر SEASON لقبول أي عنصر كتابع، ويمكن تنظيم العناصر أيـــاً كان الأمر. ويفيد هذا الأمر عندما يكون لديك نص غير ذي بنية مثل مقال بجريدة، قـــد تظـــهر فقرات ومربعات الفقرات الإضافية وقوائم التعداد النقطي والقوائم المرقمة والرســـوم والصـــور والرؤوس الفرعية في أي مكان بالمستند إلا أنك قد ترغب في بعض الأحيان في اختيـــار نظـــام وتحكم أكثر في موضع بياناتك. فعلى سبيل المثال، يمكن أن تطلب أن يكون لكل LEAGUE فقـط LEAGUE_NAME واحد، وأن يكون كـــل PLAYER لـــه GIVEN_NAME وSURNAME وأن يأتي GIVEN_NAME قبل SURNAME.

ولإعلان أن كل LEAGUE يجب أن يكون له اسم، قم فقط بإعلان عنصر LEAGUE_NAME ثم قم بتضمين LEAGUE_NAME داخل أقواس بنهاية إعلان LEAGUE كما يلي:

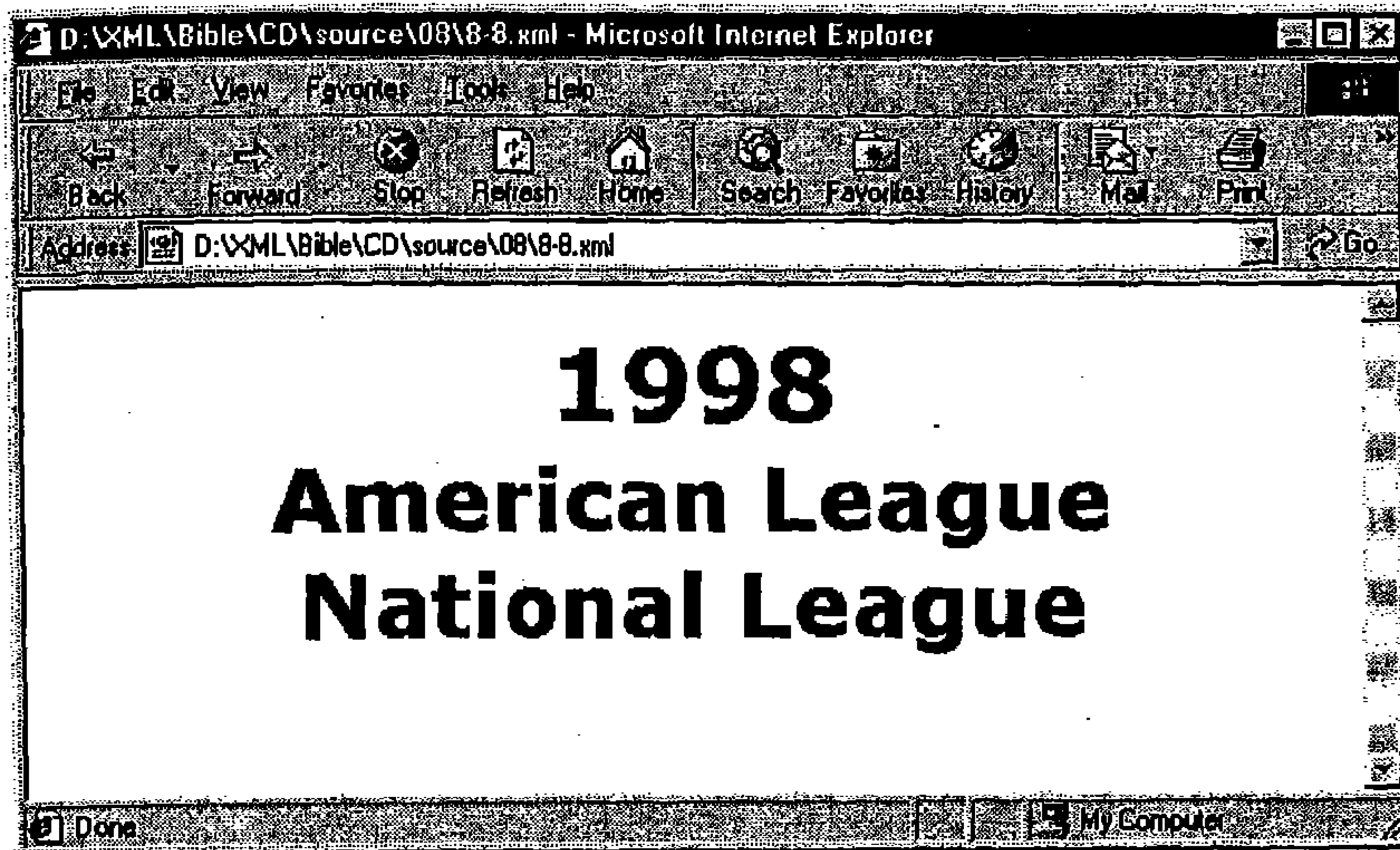
```
<!ELEMENT LEAGUE (LEAGUE_NAME)>
<!ELEMENT LEAGUE_NAME (#PCDATA)>
```

وينبغي أن يتم إعلان كل عنصر داخل إعلان <!ELEMENT> الخاص بـــه مـــرة واحـــدة بالتحديد، حتى إذا كان يظهر كتابع في إعلانات <!ELEMENT> أخرى. ولقد قمنا هنا بوضـــع إعلان LEAGUE_NAME بعد إعلان LEAGUE الذي يشير إليه، إلا أن هـــذا لا يصنـــع فارقـــاً ويسمح XML بهذه الأنواع من مراجع الإرسال الإعلامية. ويكون الترتيب الـــذي تظـــهر عليـــه علامات العناصر غير المرتبطة بالمضمون طالما أن كل الإعلانات الخاصة بها موجـــودة كلـــها داخل DTD.

ويمكن إضافة هذين الإعلانين إلى المستند ثم تضمين عنصري LEAGUE وLEAGUE_NAME في SEASON. وهذا ما توضحه تعليمات البرمجة ٨-٨. ويوضح الشكل ٨-٨ المستند الناتج.

تعليمات البرمجة ٨-٨: SEASON يحتوي على تابعي LEAGUE

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="baseballstats.css"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME)>
  <!ELEMENT LEAGUE_NAME (#PCDATA)>
  <!ELEMENT SEASON ANY>
]>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>American League</LEAGUE_NAME>
  </LEAGUE>
  <LEAGUE>
    <LEAGUE_NAME>National League</LEAGUE_NAME>
  </LEAGUE>
</SEASON>
```



الشكل ٨-٨ مستند صالح يحتوي على ورقة نمط وعنصر YEAR وتابعي LEAGUE.

التسلسلات

لتقوم أيضاً بحصر عنصر SEASON. فعنصر SEASON يحتوي على YEAR واحد بالتحديد متبوعاً بعنصري LEAGUE بالضبط وبدلاً من أن نقول أن SEASON قد يحتوي على عناصر ANY، قم بتحديد هذه التوابع الثلاثة بتضمينها بإعلان عنصر SEASON تحيطها أقواس وتفصلها فواصل كما يلي:

```
<!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
```

ويطلق على قائمة العناصر التابعة المنفصلة بفواصل "تسلسل". وبواسطة هذا الإعلان، لابد أن يحتوي كل عنصر SEASON سليم على عنصر YEAR واحد بالتحديد متبوعاً بعنصري LEAGUE ولا شيء آخر. والآن يبدو إعلان نوع المستند الكامل كما يلي:

```
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME)>
  <!ELEMENT LEAGUE_NAME (#PCDATA)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
]>
```

ويلتزم جزء المستند من تعليمات البرمجة ٨-٨ بما في DTD حيث أن عنصر SEASON الخاص به يحتوي على تابع YEAR واحد متبوعاً بتابعي LEAGUE فقط. إلا أنه إذا اشتمل

المستند على LEAGUE واحد فقط، فسيكون المستند، بالرغم من أنه صحيح التكوين، سيكون غير صالح وبالمثل، إذا أتى LEAGUE قبل عنصر YEAR وليس بعده، أو إذا احتوى عنصر LEAGUE على توابع، أو إذا لم يلتزم المستند بأي حال من الأحوال بما في DTD، فسوف يكون المستند غير صالح وسترفضه موزعات التحقق من الصحة.

ومن السهل توسيع هذه التقنيات لتغطية التقسيم. ويحتوي كل LEAGUE، كما في LEAGUE_NAME على ثلاثة توابع DIVISION. فعلى سبيل المثال:

```
<!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
DIVISION)>
```

تابع واحد أو أكثر

يحتوي كل DIVISION على DIVISION_NAME وما بين ٤ و٦ توابع TEAM. ومن السهل تعيين DIVISION_NAME. وذلك موضح بأسفل:

```
<!ELEMENT DIVISION (DIVISION_NAME)>
 <!ELEMENT DIVISION_NAME (#PCDATA)>
```

إلا أن توابع TEAM ذات إمكانات أعلى ومن السهل الإخبار بأنك تريد ٤ توابع TEAM في DIVISION، كما هو موضح بأسفل:

```
<!ELEMENT DIVISION (DIVISION_NAME, TEAM, TEAM, TEAM, TEAM)>
```

وليس من الصعب تحديد ٥ أو ٦ توابع ولكن كيف يمكنك التعبير عن رغبتك في ما بين ٤ و٦ وما بينهما؟ في الواقع لا يقدم XML طريقة سهلة للقيام بذلك. ولكن يمكنك القول أنك تريد عنصر محدد أو أكثر من عنصر بوضع علامة الجمع "+" بعد اسم العنصر في قائمة التوابع. فعلى سبيل المثال:

```
<!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
```

وتقول هذه العبارة أن عنصر DIVISION ينبغي أن يحتوي على عنصر DIVISION_NAME متبوعاً بعنصر TEAM أو أكثر.



هناك طريقة صعبة للتعبير عن رغبتك في احتواء عنصر DIVISION على ما بين ٤ و٦ عناصر TEAM، وليس ٣ أو ٧. إلا أنه سيكون من المؤسف أنه لا يمكن استخدامها عملياً. وبمجرد انتهاءك من قراءة هذا الفصل، حاول استكشاف القيام بذلك.

بدون توابع أو أكثر من تابع

ينبغي أن يحتوي كل TEAM على TEAM_CITY واحد وTEAM_NAME واحد وعـــدد غــير محدد من عناصر PLAYER. وفي الحقيقة ستحتاج على الأقل إلى تسعة لاعبين لفريق البيسـبول، إلا أننا في الأمثلة الموجودة في هذا الكتاب قمنا بإدراج العديد من الفرق بـــدون لاعبيـــن وذلـــك بسبب مشكلة المساحة لذلك فإننا نرغب في تحديد أن أي TEAM قد لا يحتوي على أيـــة توابـــع PLAYER أو على أكثر من تابعٍ. ويمكنك القيام بذلك بإلحاق علامة نجمية (*) باسم العنصر فـــي قائمة التوابع. فعلى سبيل المثال:

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
  <!ELEMENT TEAM_CITY (#PCDATA)>
  <!ELEMENT TEAM_NAME (#PCDATA)>
```

بدون توابع أو تابع واحد

تعتبر العناصر النهائية في المستند والتي يتم إشراكها داخل اللعبة هي توابـــع PLAYER. وهـــي تعتبر كلها عناصر بسيطة تحتوي على النص فقط وهذه هي الإعلانات الخاصة بها.

```
<!ELEMENT SURNAME (#PCDATA)>
  <!ELEMENT GIVEN_NAME (#PCDATA)>
  <!ELEMENT POSITION (#PCDATA)>
  <!ELEMENT GAMES (#PCDATA)>
  <!ELEMENT GAMES_STARTED (#PCDATA)>
  <!ELEMENT AT_BATS (#PCDATA)>
  <!ELEMENT RUNS (#PCDATA)>
  <!ELEMENT HITS (#PCDATA)>
  <!ELEMENT DOUBLES (#PCDATA)>
  <!ELEMENT TRIPLES (#PCDATA)>
  <!ELEMENT HOME_RUNS (#PCDATA)>
  <!ELEMENT RBI (#PCDATA)>
  <!ELEMENT STEALS (#PCDATA)>
  <!ELEMENT CAUGHT_STEALING (#PCDATA)>
  <!ELEMENT SACRIFICE_HITS (#PCDATA)>
  <!ELEMENT SACRIFICE_FLIES (#PCDATA)>
  <!ELEMENT ERRORS (#PCDATA)>
```

```
<!ELEMENT WALKS (#PCDATA)>
<!ELEMENT STRUCK_OUT (#PCDATA)>
<!ELEMENT HIT_BY_PITCH (#PCDATA)>
<!ELEMENT COMPLETE_GAMES (#PCDATA)>
<!ELEMENT SHUT_OUTS (#PCDATA)>
<!ELEMENT ERA (#PCDATA)>
<!ELEMENT INNINGS (#PCDATA)>
<!ELEMENT EARNED_RUNS (#PCDATA)>
<!ELEMENT HIT_BATTER (#PCDATA)>
<!ELEMENT WILD_PITCHES (#PCDATA)>
<!ELEMENT BALK (#PCDATA)>
<!ELEMENT WALKED_BATTER (#PCDATA)>
<!ELEMENT WINS (#PCDATA)>
<!ELEMENT LOSSES (#PCDATA)>
<!ELEMENT SAVES (#PCDATA)>
<!ELEMENT COMPLETE_GAMES (#PCDATA)>
<!ELEMENT STRUCK_OUT_BATTER (#PCDATA)>
```

والآن يمكن كتابة الإعلان لعنصر PLAYER. ولدى كل لاعب SURNAME واحد وGIVEN_NAME واحد وPOSITION واحد وGAMES واحدة. ويمكن أيضاً إعلان أن كل PLAYER أيضاً له AT_BATS وRUNS وHITS واحدة وهكذا. إلا أنه ليس من المؤكد أن من الصحيح إدراج عدد صفري لرميات رامي لم يقم بضرب الكرة ومن المحتمل أن يؤدي ذلك إلى تقسيم بأخطاء صفرية عند بدء حساب معدلات الضرب وهكذا. وإذا لم ينطبق عنصر معين على لاعب أو إذا لم يكن ذلك متاحاً، فسيكون أنسب شيء تقوم به هو تجاهل الإحصائية المعينة من المعلومات الخاصة باللاعب. ولا يسمح بأكثر من واحد من كل عنصر للاعب معين. لذلك فتحتاج إلى عنصر صفري أو عنصر واحد للنوع المقدم. ويمكن الإشارة إلى ذلك في قائمة العناصر التابعة بإلحاق علامة استفهام (؟) بالعنصر، كما هو موضح أسفل:

```
<!ELEMENT PLAYER (GIVEN_NAME, SURNAME, POSITION, GAMES,
   GAMES_STARTED, AT_BATS?, RUNS?, HITS?, DOUBLES?,
   TRIPLES?, HOME_RUNS?, RBI?, STEALS?, CAUGHT_STEALING?,
   SACRIFICE_HITS?, SACRIFICE_FLIES?, ERRORS?, WALKS?,
   STRUCK_OUT?, HIT_BY_PITCH?, WINS?, LOSSES?, SAVES?,
   COMPLETE_GAMES?, SHUT_OUTS?, ERA?, INNINGS?,
EARNED_RUNS?,
```

```
HIT_BATTER?,WILD_PITCHES?, BALK?,WALKED_BATTER?,
STRUCK_OUT_BATTER?)
>
```

ويوضح ذلك أن كل PLAYER له PLAYER وSURNAME وPOSITION GIVEN_NAME وGAMES وGAMES_STARTED. وأيضاً قد يكون لكل لاعــب AT_BATS وRUNS وHITS وDOUBLES وTRIPLES وHOME_RUNS وRBI وSTEALS وCAUGHT_STEALING وSACRIFICE_HITS وSACRIFICE_FLIES وERRORS, WALKS وSTRUCK_OUT وHIT_BY_PITCH واحد أولاً يكون لديه أي منها.

المستند الكامل وتعريف نوع المستند

لدينا الآن DTD كامل لإحصاءات البيسبول. وتوضح تعليمات البرمجة ٨-٩ DTD هذا وكذلــك الجزء الخاص بالمستند من تعليمات البرمجة ٨-٤.

تعليمات البرمجة ٨-٩: مستند XML صحيح في إحصاءات البيسبول يحتوي على DTD

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
DIVISION)>
  <!ELEMENT LEAGUE_NAME (#PCDATA)>
  <!ELEMENT DIVISION_NAME (#PCDATA)>
  <!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
  <!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
  <!ELEMENT TEAM_CITY (#PCDATA)>
  <!ELEMENT TEAM_NAME (#PCDATA)>
  <!ELEMENT PLAYER (GIVEN_NAME, SURNAME, POSITION, GAMES,
    GAMES_STARTED, WINS?, LOSSES?, SAVES?,
 AT_BATS?, RUNS?, HITS?, DOUBLES?, TRIPLES?, HOME_RUNS?,
    RBI?, STEALS?, CAUGHT_STEALING?, SACRIFICE_HITS?,
```

```
    SACRIFICE_FLIES?, ERRORS?, WALKS?, STRUCK_OUT?,
    HIT_BY_PITCH?, COMPLETE_GAMES?, SHUT_OUTS?, ERA?,
INNINGS?,
    EARNED_RUNS?, HIT_BATTER?, WILD_PITCHES?, BALK?,
    WALKED_BATTER?, STRUCK_OUT_BATTER?)
>
  <!ELEMENT SURNAME (#PCDATA)>
  <!ELEMENT GIVEN_NAME (#PCDATA)>
  <!ELEMENT POSITION (#PCDATA)>
  <!ELEMENT GAMES (#PCDATA)>
  <!ELEMENT GAMES_STARTED (#PCDATA)>
  <!ELEMENT COMPLETE_GAMES (#PCDATA)>
  <!ELEMENT WINS (#PCDATA)>
  <!ELEMENT LOSSES (#PCDATA)>
  <!ELEMENT SAVES (#PCDATA)>
  <!ELEMENT AT_BATS (#PCDATA)>
  <!ELEMENT RUNS (#PCDATA)>
  <!ELEMENT HITS (#PCDATA)>
  <!ELEMENT DOUBLES (#PCDATA)>
  <!ELEMENT TRIPLES (#PCDATA)>
  <!ELEMENT HOME_RUNS (#PCDATA)>
  <!ELEMENT RBI (#PCDATA)>
  <!ELEMENT STEALS (#PCDATA)>
  <!ELEMENT CAUGHT_STEALING (#PCDATA)>
  <!ELEMENT SACRIFICE_HITS (#PCDATA)>
  <!ELEMENT SACRIFICE_FLIES (#PCDATA)>
  <!ELEMENT ERRORS (#PCDATA)>
  <!ELEMENT WALKS (#PCDATA)>
  <!ELEMENT STRUCK_OUT (#PCDATA)>
  <!ELEMENT HIT_BY_PITCH (#PCDATA)>
  <!ELEMENT SHUT_OUTS (#PCDATA)>
  <!ELEMENT ERA (#PCDATA)>
  <!ELEMENT INNINGS (#PCDATA)>
  <!ELEMENT HOME_RUNS_AGAINST (#PCDATA)>
```

```
 <!ELEMENT RUNS_AGAINST (#PCDATA)>
 <!ELEMENT EARNED_RUNS (#PCDATA)>
 <!ELEMENT HIT_BATTER (#PCDATA)>
 <!ELEMENT WILD_PITCHES (#PCDATA)>
 <!ELEMENT BALK (#PCDATA)>
 <!ELEMENT WALKED_BATTER (#PCDATA)>
 <!ELEMENT STRUCK_OUT_BATTER (#PCDATA)>

]>
<SEASON>
 <YEAR>1998</YEAR>
 <LEAGUE>
  <LEAGUE_NAME>National</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Florida</TEAM_CITY>
      <TEAM_NAME>Marlins</TEAM_NAME>
      <PLAYER>
        <GIVEN_NAME>Eric</GIVEN_NAME>
        <SURNAME>Ludwick</SURNAME>
        <POSITION>Starting Pitcher</POSITION>
        <GAMES>13</GAMES>
        <GAMES_STARTED>6</GAMES_STARTED>
        <WINS>1</WINS>
        <LOSSES>4</LOSSES>
        <SAVES>0</SAVES>
        <COMPLETE_GAMES>0</COMPLETE_GAMES>
        <SHUT_OUTS>0</SHUT_OUTS>
        <ERA>7.44</ERA>
        <INNINGS>32.2</INNINGS>
        <EARNED_RUNS>31</EARNED_RUNS>
        <HIT_BATTER>27</HIT_BATTER>
        <WILD_PITCHES>0</WILD_PITCHES>
```

```
        <BALK>2</BALK>
        <WALKED_BATTER>0</WALKED_BATTER>
        <STRUCK_OUT_BATTER>17</STRUCK_OUT_BATTER>
      </PLAYER>
      <PLAYER>
        <GIVEN_NAME>Brian</GIVEN_NAME>
        <SURNAME>Daubach</SURNAME>
        <POSITION>First Base</POSITION>
        <GAMES>10</GAMES>
        <GAMES_STARTED>3</GAMES_STARTED>
        <AT_BATS>15</AT_BATS>
        <RUNS>0</RUNS>
        <HITS>3</HITS>
        <DOUBLES>1</DOUBLES>
        <TRIPLES>0</TRIPLES>
        <HOME_RUNS>0</HOME_RUNS>
        <RBI>3</RBI>
        <STEALS>0</STEALS>
        <CAUGHT_STEALING>0</CAUGHT_STEALING>
        <SACRIFICE_HITS>0</SACRIFICE_HITS>
        <SACRIFICE_FLIES>0</SACRIFICE_FLIES>
        <ERRORS>0</ERRORS>
        <WALKS>1</WALKS>
        <STRUCK_OUT>5</STRUCK_OUT>
        <HIT_BY_PITCH>1</HIT_BY_PITCH>
      </PLAYER>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Montreal</TEAM_CITY>
      <TEAM_NAME>Expos</TEAM_NAME>
</TEAM>
    <TEAM>
      <TEAM_CITY>New York</TEAM_CITY>
      <TEAM_NAME>Mets</TEAM_NAME>
```

```
    </TEAM>
    <TEAM>
      <TEAM_CITY>Philadelphia</TEAM_CITY>
    <TEAM_NAME>Phillies</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Chicago</TEAM_CITY>
      <TEAM_NAME>Cubs</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Arizona</TEAM_CITY>
      <TEAM_NAME>Diamondbacks</TEAM_NAME>
    </TEAM>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Baltimore</TEAM_CITY>
      <TEAM_NAME>Orioles</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Chicago</TEAM_CITY>
```

```
      <TEAM_NAME>White Sox</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Anaheim</TEAM_CITY>
      <TEAM_NAME>Angels</TEAM_NAME>
    </TEAM>
  </DIVISION>
 </LEAGUE>
</SEASON>
```

ولا تعتبر تعليمات البرمجة ٨-٩ هي المستند الوحيد المتاح الذي يطابق DTD هذا. فهناك تعليمات البرمجة ٨-١٠ التي تعتبر أيضاً مستند صحيح، حيث أنه يحتوي على كل العناصر المطلوبة في الترتيب المطلوب ولا يحتوي على أية عناصر غير معلنة. ويعتبر هذا المستند هو أقل المستندات المناسبة والتي يمكن إنشاؤها وتلائم DTD. والعوامل المحددة لهذا الأمر هي ضرورة احتواء كل SEASON على تابعي LEAGUE واحتواء كل LEAGUE على ثلاثة توابع DIVISION واحتواء كل DIVISION على الأقل على TEAM واحد.

تعليمات البرمجة ٨-١٠: مستند XML آخر صحيح بحسب DTD للبيسبول

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
DIVISION)>
  <!ELEMENT LEAGUE_NAME (#PCDATA)>
  <!ELEMENT DIVISION_NAME (#PCDATA)>
  <!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
  <!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
  <!ELEMENT TEAM_CITY (#PCDATA)>
```

```
<!ELEMENT TEAM_NAME (#PCDATA)>
<!ELEMENT PLAYER (GIVEN_NAME, SURNAME, POSITION, GAMES,
  GAMES_STARTED, COMPLETE_GAMES?, WINS?, LOSSES?, SAVES?,
  AT_BATS?, RUNS?, HITS?, DOUBLES?, TRIPLES?, HOME_RUNS?,
  RBI?, STEALS?, CAUGHT_STEALING?, SACRIFICE_HITS?,
  SACRIFICE_FLIES?, ERRORS?, WALKS?, STRUCK_OUT?,
  HIT_BY_PITCH?, COMPLETE_GAMES?, SHUT_OUTS?, ERA?,
INNINGS?,
  EARNED_RUNS?, HIT_BATTER?, WILD_PITCHES?, BALK?,
  WALKED_BATTER?, STRUCK_OUT_BATTER?)
>
<!ELEMENT SURNAME (#PCDATA)>
<!ELEMENT GIVEN_NAME (#PCDATA)>
<!ELEMENT POSITION (#PCDATA)>
<!ELEMENT GAMES (#PCDATA)>
<!ELEMENT GAMES_STARTED (#PCDATA)>
<!ELEMENT COMPLETE_GAMES (#PCDATA)>
<!ELEMENT WINS (#PCDATA)>
<!ELEMENT LOSSES (#PCDATA)>
<!ELEMENT SAVES (#PCDATA)>
<!ELEMENT AT_BATS (#PCDATA)>
<!ELEMENT RUNS (#PCDATA)>
<!ELEMENT HITS (#PCDATA)>
<!ELEMENT DOUBLES (#PCDATA)>
<!ELEMENT TRIPLES (#PCDATA)>
<!ELEMENT HOME_RUNS (#PCDATA)>
<!ELEMENT RBI (#PCDATA)>
<!ELEMENT STEALS (#PCDATA)>
<!ELEMENT CAUGHT_STEALING (#PCDATA)>
<!ELEMENT SACRIFICE_HITS (#PCDATA)>
<!ELEMENT SACRIFICE_FLIES (#PCDATA)>
<!ELEMENT ERRORS (#PCDATA)>
<!ELEMENT WALKS (#PCDATA)>
<!ELEMENT STRUCK_OUT (#PCDATA)>
```

```
  <!ELEMENT HIT_BY_PITCH (#PCDATA)>
  <!ELEMENT SHUT_OUTS (#PCDATA)>
  <!ELEMENT ERA (#PCDATA)>
  <!ELEMENT INNINGS (#PCDATA)>
  <!ELEMENT HOME_RUNS_AGAINST (#PCDATA)>
  <!ELEMENT RUNS_AGAINST (#PCDATA)>
  <!ELEMENT EARNED_RUNS (#PCDATA)>
  <!ELEMENT HIT_BATTER (#PCDATA)>
  <!ELEMENT WILD_PITCHES (#PCDATA)>
  <!ELEMENT BALK (#PCDATA)>
  <!ELEMENT WALKED_BATTER (#PCDATA)>
  <!ELEMENT STRUCK_OUT_BATTER (#PCDATA)>

]>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>National</LEAGUE_NAME>
    <DIVISION>
      <DIVISION_NAME>East</DIVISION_NAME>
        <TEAM>
          <TEAM_CITY>Atlanta</TEAM_CITY>
          <TEAM_NAME>Braves</TEAM_NAME>
        </TEAM>
        <TEAM>
          <TEAM_CITY>Florida</TEAM_CITY>
          <TEAM_NAME>Marlins</TEAM_NAME>
        </TEAM>
        <TEAM>
          <TEAM_CITY>Montreal</TEAM_CITY>
          <TEAM_NAME>Expos</TEAM_NAME>
        </TEAM>
        <TEAM>
          <TEAM_CITY>New York</TEAM_CITY>
```

```
      <TEAM_NAME>Mets</TEAM_NAME>
    </TEAM>
    <TEAM>
      <TEAM_CITY>Philadelphia</TEAM_CITY>
    <TEAM_NAME>Phillies</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Chicago</TEAM_CITY>
      <TEAM_NAME>Cubs</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Arizona</TEAM_CITY>
      <TEAM_NAME>Diamondbacks</TEAM_NAME>
     </TEAM>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Baltimore</TEAM_CITY>
      <TEAM_NAME>Orioles</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
     <TEAM>
```

```
      <TEAM_CITY>Chicago</TEAM_CITY>
      <TEAM_NAME>White Sox</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Anaheim</TEAM_CITY>
      <TEAM_NAME>Angels</TEAM_NAME>
    </TEAM>
  </DIVISION>
</LEAGUE>
</SEASON>
```

الاختيارات

يحتوي العنصر الأصلي الواحد بصفة عامة على توابع عديدة. ويتم فصل هذه التوابع بفواصل، للإشارة إلى ضرورة تواجد التوابع في تسلسل. إلا أن مثل هذا العنصر التابع قد يتم إلحاقه بعلامة استفهام أو علامة الجمع أو علامة نجمية وذلك لضبط عدد مرات ظهوره في هذا الموقع من التسلسل.

وحتى الآن، فإن الافتراض قام على أن العناصر التابعة تظهر أو لا تظهر بترتيب معين. إلا أنك، قد ترغب في جعل DTD أكثر مرونة في موضع محدد. فعلى سبيل المثال، لابد أن يكون لكل عنصر PAYMENT، في DTD الذي يصف البيع على حسب العملاء، إما تابع CREDIT_CARD أو تابع CASH مقدماً معلومات عن طريقة الدفع. إلا أن أي PAYMENT فردي لن يكون له أي منهما.

ويمكن الإشارة إلى أن منشئ المستند في حاجة إلى إدخال إما عنصر أو آخر لفصل العناصر التابعة بشريط عمودي (|) بدلاً من فاصلة (,) في إعلان العنصر الأصلي. فعلى سبيل المثال، يوضح ما يلي أن عنصر PAYMENT ينبغي أن يحتوي على تابع واحد من نوع CASH أو CREDIT_CARD.

```
<!ELEMENT PAYMENT (CASH | CREDIT_CARD)>
```

ويطلق على هذا الفرز لمواصفة المحتوى "اختيار" ويمكن فصل أي عدد من التوابع بأشرطة عمودية عند الحاجة إلى استخدام أحدها بالتحديد. فعلى سبيل المثال، يوضح ما يلي أن عنصر PAYMENT ينبغي أن يكون له تابع لنوع CASH أو CREDIT_CARD أو CHECK.

```
<!ELEMENT PAYMENT (CASH | CREDIT_CARD | CHECK)>
```

ويكون الشريط العمودي أكثر فائدة عند تجميع العناصر باستخدام أقواس. ويمكن تجميع مركبات من العناصر داخل أقواس ثم إلحاق الأقواس بعلامات نجمية أو علامات استفهام وعلامات جمع للإشارة إلى ضرورة تواجد مركبات معينة من العناصر بعدد مرات صفر أو أكثر، أو صفر أو مرة واحدة، أو مرة واحدة أو أكثر.

تجميع التوابع داخل أقواس

آخر ما تحتاج معرفته عن ترتيب العناصر التابعة في إعلانات العناصر الأصل هو كيفية تجميع العناصر بواسطة أقواس وتجمع كل مجموعة من الأقواس بين عناصر عديدة كعنصر واحد. وعندئذ يمكن تضمين هذا العنصر، والموجود داخل قوسين، داخل أقواس أخرى في موقع عنصر فردي. وقد يحتوي أيضاً على علامة الجمع أو فاصلة أو علامة استفهام ملحقة به. ويمكن تجميع هذه المركبات الموجودة داخل أقواس في مجموعات أكبر مضمنة داخل أقواس، لإنتاج بنية مركبة. وهي تعتبر تقنية ذات فاعلية.

فعلى سبيل المثال، لنعتبر أن هناك قائمة تتكون من عنصرين يجب تبديل كل منهما بالآخر. وهذه هي بالضرورة الطريقة التي يعمل بها تعريف HTML. فكل علامة <dt> يجب أن تطابق إحدى علامات <dd>. وعند مماثلة هذه البنية في XML، سيبدو إعلان عنصر dl كما يلي:

```
<!ELEMENT dl (dt, dd)*>
```

وتشير الأقواس إلى تكرار <dt><dd> المطابقين، وليس <dd> فقط.

وغالباً ما تظهر العناصر بترتيب أكثر أو أقل عشوائية وبصفة عامة فإن مقالات مجلات الأخبار يكون لها عنوان متبوعاً بفقرات نصية تحتوي على رسوم وصور مربعات الفقرات الإضافية والرؤوس الفرعية واستشهادات منشورة وقد تنتهي بسطر توقيع فيمكنك الإشارة إلى هذا النوع من الترتيب بسرد كل العناصر التابعة المتاحة في إعلان العنصر الأصل تفصلها شرائط عمودية ومجمعة داخل أقواس ويمكنك عندئذ وضع علامة نجمية خارج قوس الإغلاق للإشارة إلى وجود عدد، قد يكون صفر أو أكثر، من العناصر المتاحة داخل الأقواس. فعلى سبيل المثال:

```
<!ELEMENT ARTICLE (TITLE, (P | PHOTO | GRAPH | SIDEBAR
 | PULLQUOTE | SUBHEAD)*, BYLINE?)>
```

وكمثال آخر، بفرض أنك تريد توضيح أن عنصر DOCUMENT، بدلاً من احتوائه على توابع، يجب أن يكون له TITLE واحد متبوعاً بأي عدد من الفقرات النصية والصور التي يمكن مزجها بحرية، متبوعة بقالب SIGNATURE خيارية. فقم بكتابة إعلان العنصر الخاص به كما يلي:

```
<!ELEMENT DOCUMENT (TITLE, (PARAGRAPH | IMAGE)*,
SIGNATURE?)>
```

وهذه ليست الطريقة الوحيدة لوصف هذه البنية. وفي الواقع قد لا تكون حتى أفضل الطـــرق. فمن الخيارات الأخرى أن تقوم بإعلان عنصر BODY الـــذي يحتـــوي علـــى PARAGRAPH وعناصر IMAGE، وتضمين ذلك بين TITLE وSIGNATURE . فعلى سبيل المثال:

```
<!ELEMENT DOCUMENT (TITLE, BODY, SIGNATURE?)>
<!ELEMENT BODY ((PARAGRAPH | IMAGE)*)>
```

والفارق بين الطريقتين هو أن الطريقة الثانية تتطلب عنصر BODY آخر من المستند. وهـــذا العنصر يقدم مستوى إضافي للترتيب الذي قد يكون "أو قد لا يكون" مفيداً للتطبيق الـــذي يقـــوم بقراءة المستند والسؤال الآن هو ما إذا كان قارئ هذا المستند "والذي قد يكون برنامج كمبيوتـــر آخر" قد يرغب في اعتبار BODY عنصر فردي منفصل عن TITLE وSIGNATURE مكتسب من كمية عناصره.

وكمثال آخر، انظر إلى العناوين الدولية الموضحة. ولا تتبع العناوين خارج الولايات المتحـدة عرف U.S. وبصفة خاصة، فإن الرموز البريدية تسبق الدولة في بعض الأحيان أو تتبع البلد كما في المثالين التاليين:

Doberman-YPPAN
Box 2021
St. Nicholas QUEBEC
CAN GOS-3LO

أو

Editions Sybex
10/12 Villa Coeur-de-Vey
75685 Paris Cedex 14
France

وبالرغم من أن بريدك قد يصل حتى إذا كانت بعض أجزاء العنوان خارج الـــترتيب، فمـــن الأفضل جعل العنوان أكثر مرونة. وإليك إعلان عنصر عنوان يتيح ذلك:

```
<!ELEMENT ADDRESS (STREET+, (CITY | STATE | POSTAL_CODE
  | COUNTRY)*)>
```

وهذا يوضح أن عنصر ADDRESS يجب أن يكون له تابع STREET واحد أو أكثر متبوعـــاً بأي عدد من عناصر CITY أو STATE أو POSTAL_CODE أو COUNTRY. وحتـــى ذلـــك يعتبر أقل مثالية إذا كنت ترغب في إتاحة ما لا يزيد على واحد لكل منها. ولسوء الحظ فإن ذلـــك

يعتبر خارج نطاق التنفيذ في DTD. فبالسماح بترتيب أكثر مرونة للعناصر، يتم إلغاء بعض إمكانيات التحكم في العدد الأقصى لكل عنصر.

ومن ناحية أخرى، قد يكون لديك قائمة تتكون من أنواع مختلفة من العناصر والتي قد تظهر بترتيب عشوائي كما في قائمة السجلات التي قد تحتوي على اسطوانات أو صور فوتوغرافية "ألبومات" أو شرائط، فسيبدو إعلان العنصر للتفرقة بين الفئات المختلفة لهذه القائمة كما يلي:

```
<!ELEMENT MUSIC_LIST (CD | ALBUM | TAPE)*>
```

ويمكن استخدام الأقواس الموجودة في DTD للبيسبول لتحديد مجموعات الإحصاءات المختلفة للرماه والمهاجمين وقد يكون لكل لاعب مجموعة واحدة أو مجموعة أخرى ولكن ليس مجموعتين. ويبدو إعلان العنصر كما يلي:

```
<!ELEMENT PLAYER (GIVEN_NAME, SURNAME, POSITION, GAMES,
   GAMES_STARTED, (( COMPLETE_GAMES?, WINS?, LOSSES?,
SAVES?,
   SHUT_OUTS?, ERA?, INNINGS?, EARNED_RUNS?, HIT_BATTER?,
   WILD_PITCHES?, BALK?, WALKED_BATTER?,
STRUCK_OUT_BATTER? )
 |(AT_BATS?, RUNS?, HITS?, DOUBLES?, TRIPLES?, HOME_RUNS?,
   RBI?, STEALS?, CAUGHT_STEALING?, SACRIFICE_HITS?,
   SACRIFICE_FLIES?, ERRORS?, WALKS?, STRUCK_OUT?,
   HIT_BY_PITCH? )))>
```

ولا تزال هناك بعض الأشياء التي يصعب تناولها في إعلانات العناصر. فعلى سبيل المثال، لا توجد طريقة جيدة لتوضيح أن المستند يجب أن يبدأ بعنصر TITLE وينتهي بعنصر SIGNATURE إلا أنه قد يحتوي على عناصر أخرى بين هذين العنصرين. وذلك حيث أن ANY قد لا ينضم إلى عناصر أخرى تابعة.

وبصفة عامة كلما قلت الدقة والإحكام فيما يتعلق بأماكن ظهور الأشياء، تقل إمكانية التحكم في عدد ما تريد ظهوره. فعلى سبيل المثال، لا يمكن توضيح أن المستند ينبغي أن يكون له عنصر TITLE واحد بالتحديد ولكن TITLE قد يظهر في أي مكان بالمستند.

وبالرغم من ذلك، فإن استخدام الأقواس في إنشاء قوالب للعناصر، إما بتسلسل أو فاصلة أو بموازاة شريط عمودي، يتيح لك إنشاء بنيات معقدة تحتوي على قواعد تفصيلية عن كيفية تتبع العناصر المختلفة للعناصر الأخرى. فحاول، بالرغم من ذلك، عدم تجاوز هذا. ودائماً ما تكون

الحلول الأكثر بساطة هي الأفضل. فكلما كان DTD أكثر تعقيداً، كلما كان من الصعب عليه كتابة ملفات صحيحة تفي بشروط DTD لعدم توضيح أي شيء عن صعوبة صيانة DTD نفسه.

المحتوى المركب

قد تكون لاحظت في معظم الأمثلة الموضحة حتى الآن أن العناصر إما أن تحتوي على عناصر تابعة أو بيانات أحرف موزعة ولكنها لا تحتوي على كلاهما. وقد كان الاستثناء الوحيد فقط هو العناصر الجذرية في الأمثلة السابقة عندما لم يكن قد تم تطوير قائمة العلامات الكاملة. ففي هذه الحالات، وحيث أن العنصر الجذري قد يحتوي بيانات ANY، فقد كان من المسموح تضمين كل من العناصر التابعة والنص الخالص.

ويمكن إعلان علامات تحتوي على كل من العناصر التابعة وبيانات الأحرف الموزعة. وهذا ما يطلق عليه mixed content أو المحتوى المختلط أو المركب. ويمكنك استخدامه لإتاحة قالب عشوائي للنص ليكون ملحقاً بكل TEAM. فعلى سبيل المثال:

```
<!ELEMENT TEAM (#PCDATA | TEAM_CITY | TEAM_NAME | PLAYER)*>
```

ويقيد خلط العناصر التابعة ببيانات الأحرف الموزعة البنية التي يمكن فرضها في المستندات. وعلى وجه الخصوص يمكن تحديد فقط أسماء العناصر التي يمكن ظهورها. ولا يمكن تقييد الترتيب الذي تظهر عليه أو عدد كل ما يظهر أو الظهور نفسه. وبلغة DTD، فإن المعنى الذي يجب أن يظهر عليه الجزء الخاص التابع في DTD هو كما يلي:

```
<!ELEMENT PARENT (#PCDATA | CHILD1 | CHILD2 | CHILD3 )* >
```

ويعتبر أي شيء أخر، فيما عدا تغيير عدد التوابع، تقريباً غير صحيح. فلا يمكنك استخدام فواصل أو علامات استفهام أو علامات الجمع في إعلان العنصر الذي يشتمل على PCDATA#. ومن الصحيح أن تفصل شرائط عمودية بين قائمة العناصر وPCDATA#. وأي استخدام أخر لا يعتبر صحيحاً. فعلى سبيل المثال، لا يعتبر ما يلي صحيحاً:

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*, #PCDATA)>
```

ويعتبر السبب الأساسي لتركيب المحتوى هو عندما تكون في أثناء عملية تحويل بيانات نص قديم إلى XML واختبار DTD الخاص بك بالتحقق من صحته عند إضافة علامات جديدة بدلاً من إنهاء التحويل الكامل ثم محاولة البحث عن الشوائب "أو الأخطاء". وتعتبر هذه تقنية جيدة وينصح باستخدامها وبعد كل شيء، فمن الأسهل التعرف على الخطأ في التعليمات البرمجية فوراً بعد الانتهاء منها بدلاً من إضاعة ساعات عديدة فيما بعد إلا أن هذه تعتبر فقط دعامة لك لاستخدامها عند التطوير. وليس من المفترض أن تكون ظاهرة للمستخدم. وعند انتهاءك من DTD لا ينبغي أن يخلط توابع العناصر أو العناصر التابعة ببيانات الأحرف الموزعة. ويمكنك دائماً إنشاء علامة جديدة تحتوي على بيانات أحرف موزعة.

فعلى سبيل المثال، يمكنك تضمين قالب للنص بنهاية كـل عنصـر TEAM وذلـك بـإعلان BLURB جديد يحتوي فقط على PCDATA# وإضافته كالعنصر التابع الأخير لعنصـر TEAM. وهو يبدو كما يلي:

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*, BLURB)>
<!ELEMENT BLURB (#PCDATA)>
```

وليس بالضرورة أن يقوم ذلك بتغيير نص المستند فكل ما يفعله هو إضافـة عنصـر واحـد خياري آخر لعلامتي الفتح والإغلاق الخاصتين به لكل عنصر TEAM. إلا أن ذلك يجعل المستند أكثر تعقيداً. وأيضاً، فإن تطبيقات XML التي تتلقى الشجرة من معالج XML بها وقت أكثر سهولة تعالج فيه البيانات عندما تكون داخل تنسيق موجود ببنية أكثر فاعلية متاح عن طريـق محتـوى غير مختلط.

العناصر الفارغة

كما سبق توضيح ذلك، فإنه يكون من المفيد أحياناً تعريف عنصر ليس به محتوى وتشتمل الأمثلة في HTML على صور وقواعد أفقية وفـاصل <IMG> و<HR> و
 وفـي XML يتـم تعريف مثل هذه العناصر الفارغة باستخدام علامات فارغة تنتهي بعلامـة /> مثـل <IMG/> و<HR/> و
.

ولابد أن تعلن المستندات الصحيحة عن كل من العناصر الفارغة وغير الفارغة المسـتخدمة. وحيث أن العناصر الفارغة على حسب التعريف ليس لها توابع، فمن السـهل الإعـلان عنـها. ويمكنك استخدام إعلان <!ELEMENT> الذي يحتوي على اسم العنصر الفارغ كالعادة، ولكـن استخدم كلمة EMPTY الأساسية (تحسن لحالة الأحرف كما في كل علامات XML) بدلاً من قائمة التوابع. فعلى سبيل المثال:

```
<!ELEMENT BR EMPTY>
<!ELEMENT IMG EMPTY>
<!ELEMENT HR EMPTY>
```

وتعتبر تعليمات البرمجة ٨-١١ مستند صحيح يستخدم كل من العنـاصر الفارغـة وغـير الفارغة.

تعليمات البرمجة ٨-١١: SEASON يحتوي على تابعي LEAGUE

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT (TITLE, SIGNATURE)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT COPYRIGHT (#PCDATA)>
  <!ELEMENT EMAIL (#PCDATA)>

T BR EMPTY>

T HR EMPTY>
  <!ELEMENT
 (#PCDATA)>
  øURE (HR, COPYRIGHT, BR, EMAIL,
     BR, LAST_MODIFIED)>
]>
<DOCUMENT>
 <TITLE>Empty Tags</TITLE>
 <SIGNATURE>
  <HR/>
  <COPYRIGHT>1999 Elliotte Rusty Harold</COPYRIGHT><BR/>
  <EMAIL>elharo@metalab.unc.edu</EMAIL><BR/>
  <LAST_MODIFIED>Thursday, April 22, 1999</LAST_MODIFIED>
 </SIGNATURE>
</DOCUMENT>
Comments in DTDs
```

تضمين تعليقات بتعريفات نوع المستند

يمكن أن تحتوي DTDs على تعليقات مثل بقية مستند XML ولا يمكن ظهور هذه التعليقات داخل إعلان، ولكن يمكن ظهورها خارجه. وغالباً ما تستخدم التعليقات لترتيب DTD في أقسام مختلفة ولتوثيق المحتوى المسموح به لعناصر معينة ولتوضيح ماهية العنصر بصورة أكثر تفصيلاً. فعلى سبيل المثال، قد يحتوي إعلان العنصر لعنصر YEAR تعليقاً يشبه الموضح أسفل:

```
<!-- A four digit year like 1998, 1999, or 2000 -->
 <!ELEMENT YEAR (#PCDATA)>
```

وكما في كل التعليقات، فإن هذا الأمر يفيد فقط الأشخاص الذين يقرؤون التعليمات البرمجية المصدر. وسوف تتجاهلها معالجات XML.

ومن الاستخدامات الممكنة للتعليقات هي تعريف الاختصارات المستخدمة في الترميز. فعلى سبيل المثال، في هذا الفصل وفي الفصول السابقة، تجنبنا استخدام الاختصارات في مصطلحات البيسبول حيث أنها ببساطة ليست واضحة كما هو متعارف عليه. وهناك طريقة أخرى بديلة وهي استخدام الاختصارات ولكن تعريفها باستخدام تعليقات في DTD.

تعليمات البرمجة ٨-١٢: مستند XML صحيح يستخدم علامات مختصرة معرفة داخل تعليقات DTD

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON [

  <!ELEMENT YEAR (#PCDATA)>
 <!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
 DIVISION)>

  <!-- American or National -->
  <!ELEMENT LEAGUE_NAME (#PCDATA)>

  <!-- East, West, or Central -->
  <!ELEMENT DIVISION_NAME (#PCDATA)>
  <!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
  <!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
  <!ELEMENT TEAM_CITY (#PCDATA)>
  <!ELEMENT TEAM_NAME (#PCDATA)>
  <!ELEMENT PLAYER (GIVEN_NAME, SURNAME, P, G,
   GS, AB?, R?, H?, D?, T?, HR?, RBI?, SB?, CS?,
   SH?, SF?, E?, BB?, S?, HBP?, CG?, SO?, ERA?, IP?,
```

```
  HRA?, RA?, ER?, HB?, WP?, B?, WB?, K?)
 >

<!- ======================== ->
 <!- Player Info ->
 <!- Player's last name ->
 <!ELEMENT SURNAME (#PCDATA)>

 <!- Player's first name ->
 <!ELEMENT GIVEN_NAME (#PCDATA)>

 <!- Position ->
 <!ELEMENT P (#PCDATA)>

 <!-Games Played ->
 <!ELEMENT G (#PCDATA)>

 <!-Games Started ->
 <!ELEMENT GS (#PCDATA)>


<!- ======================== ->
 <!- Batting Statistics ->
 <!- At Bats ->
 <!ELEMENT AB (#PCDATA)>

 <!- Runs ->
 <!ELEMENT R (#PCDATA)>

 <!- Hits ->
 <!ELEMENT H (#PCDATA)>

 <!- Doubles ->
 <!ELEMENT D (#PCDATA)>
```

```
<!- Triples ->
<!ELEMENT T (#PCDATA)>

<!- Home Runs ->
<!ELEMENT HR (#PCDATA)>

<!- Runs Batted In ->
<!ELEMENT RBI (#PCDATA)>

<!- Stolen Bases ->
<!ELEMENT SB (#PCDATA)>

<!- Caught Stealing ->
<!ELEMENT CS (#PCDATA)>

<!- Sacrifice Hits ->
<!ELEMENT SH (#PCDATA)>

<!- Sacrifice Flies ->
<!ELEMENT SF (#PCDATA)>

<!- Errors ->
<!ELEMENT E (#PCDATA)>

<!- Walks (Base on Balls) ->
<!ELEMENT BB (#PCDATA)>

<!- Struck Out ->
<!ELEMENT S (#PCDATA)>

<!- Hit By Pitch ->
<!ELEMENT HBP (#PCDATA)>
```

```
<!- ======================== ->
  <!- Pitching Statistics ->
  <!- Complete Games ->
  <!ELEMENT CG (#PCDATA)>

  <!- Shut Outs ->
  <!ELEMENT SO (#PCDATA)>

  <!- ERA ->
  <!ELEMENT ERA (#PCDATA)>

  <!- Innings Pitched ->
  <!ELEMENT IP (#PCDATA)>

<!- Home Runs hit Against ->
  <!ELEMENT HRA (#PCDATA)>

  <!- Runs hit Against ->
  <!ELEMENT RA (#PCDATA)>

  <! Earned Runs ->
  <!ELEMENT ER (#PCDATA)>

  <!- Hit Batter ->
  <!ELEMENT HB (#PCDATA)>

  <!- Wild Pitches ->
  <!ELEMENT WP (#PCDATA)>

  <!- Balk ->
  <!ELEMENT B (#PCDATA)>

  <!- Walked Batter ->
  <!ELEMENT WB (#PCDATA)>
```

```
 <!- Struck Out Batter ->
 <!ELEMENT K (#PCDATA)>

<!- ======================== ->
 <!- Fielding Statistics ->
 <!- Not yet supported ->

]>
<SEASON>
 <YEAR>1998</YEAR>
 <LEAGUE>
  <LEAGUE_NAME>National</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Atlanta</TEAM_CITY>
      <TEAM_NAME>Braves</TEAM_NAME>
      <PLAYER>
       <GIVEN_NAME>Ozzie</GIVEN_NAME>
       <SURNAME>Guillen</SURNAME>
       <P>Shortstop</P>
       <G>83</G>
       <GS>59</GS>
       <AB>264</AB>
       <R>35</R>
       <H>73</H>
       <D>15</D>
       <T>1</T>
       <HR>1</HR>
       <RBI>22</RBI>
       <SB>1</SB>
       <CS>4</CS>
       <SH>4</SH>
```

```
      <SF>2</SF>
      <E>6</E>
      <BB>24</BB>
      <S>25</S>
      <HBP>1</HBP>
    </PLAYER>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Florida</TEAM_CITY>
    <TEAM_NAME>Marlins</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Montreal</TEAM_CITY>
    <TEAM_NAME>Expos</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>New York</TEAM_CITY>
    <TEAM_NAME>Mets</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Philadelphia</TEAM_CITY>
  <TEAM_NAME>Phillies</TEAM_NAME>
  </TEAM>
</DIVISION>
<DIVISION>
  <DIVISION_NAME>Central</DIVISION_NAME>
   <TEAM>
    <TEAM_CITY>Chicago</TEAM_CITY>
    <TEAM_NAME>Cubs</TEAM_NAME>
   </TEAM>
</DIVISION>
<DIVISION>
  <DIVISION_NAME>West</DIVISION_NAME>
   <TEAM>
```

```
          <TEAM_CITY>Arizona</TEAM_CITY>
          <TEAM_NAME>Diamondbacks</TEAM_NAME>
        </TEAM>
      </DIVISION>
    </LEAGUE>
    <LEAGUE>
      <LEAGUE_NAME>American</LEAGUE_NAME>
      <DIVISION>
  <DIVISION_NAME>East</DIVISION_NAME>
        <TEAM>
          <TEAM_CITY>Baltimore</TEAM_CITY>
          <TEAM_NAME>Orioles</TEAM_NAME>
        </TEAM>
      </DIVISION>
      <DIVISION>
        <DIVISION_NAME>Central</DIVISION_NAME>
        <TEAM>
          <TEAM_CITY>Chicago</TEAM_CITY>
          <TEAM_NAME>White Sox</TEAM_NAME>
        </TEAM>
      </DIVISION>
      <DIVISION>
        <DIVISION_NAME>West</DIVISION_NAME>
        <TEAM>
          <TEAM_CITY>Anaheim</TEAM_CITY>
          <TEAM_NAME>Angels</TEAM_NAME>
        </TEAM>
      </DIVISION>
    </LEAGUE>
  </SEASON>
```

وعند تضمين Major League كاملاً يقل المستند الناتج من ٦٩٩ كيلو مع العلامات الطويلة إلى ٣٩١ كيلو مع علامات قصيرة ونسبة حفظ تصل إلى ٤٤ بالمائة. ويكون ظاهراً أن محتوى المعلومات هو نفسه على الرغم من ذلك وبناءاً على ذلك تكون الأحجام المضغوطة لمستندين

أكثر تقارباً ٥٨ كيلو للمستند الذي يحتوي على علامات قصيرة، في مقابل ٦٦ كيلو للمستند الذي يحتوي على علامات طويلة.

وليس هناك حد معين لكمية المعلومات التي يمكن أو ينبغي تضمينها في تعليقات. وبتضمين كم أكبر، يصبح DTD أكثر طولاً "ولذلك يكون المسح الضوئي أكثر صعوبة والتحميل أبطأ". إلا أنك ستتعرف في الفصلين التاليين على طرق لإعادة استخدام DTD نفسه في مستندات XML متعددة، وأيضاً تقسيم DTDs طويلة إلى أجزاء أبسط. لذلك فإن عيوب استخدام التعليقات يعتبر مؤقتة. وننصحك باستخدام التعليقات بحرية في كل DTDs. ولكن بصفة خاصة في تلك المتاحة للاستخدام العام.

مشاركة تعريفات نوع المستند الشائعة بين المستندات

لقد تضمنت الأمثلة الصحيحة السابقة DTD في تسجيل المستند الأولي. إلا أن قوة XML الحقيقية تأتي من DTDs الشائعة التي يمكن إشراكها بين مستندات عديدة تمت كتابتها عن طريق العديد من الأشخاص. وإذا لم يتم تضمين DTD مباشرة بالمستند ولكن تم ربطه من مصدر خارجي فإن التغييرات التي يتم إجرائها على DTD تنتقل آلياً إلى كل المستندات باستخدام DTD. ومن جانب آخر، لا يكون التوافق الخلفي مضمون إذا كان DTD قد تم تعديله وقد يحدث قطع المستندات بسبب التغييرات غير المتوافقة.

وعند استخدام DTD خارجي، يتغير إعلان نوع المستند، وبدلاً من تضمين DTD داخل قوسين مربعين، ويتبع كلمة SYSTEM الأساسية URL مطلق أو نسبي حيث قد يتواجد DTD، فعلى سبيل المثال:

```
<!DOCTYPE root_element_name SYSTEM "DTD_URL">
```

وفي هذا المثال root_element_name هو فقط اسم العنصر الجذري، وSYSTEM هو كلمة XML الأساسية وDTD_URL هو URL نسبي أو مطلق حيث قد يتواجد DTD. فعلى سبيل المثال:

```
<!DOCTYPE SEASON SYSTEM "baseball.dtd">
```

ولننتقل الآن لمثال معروف يوضح لنا هذه العملية. فتتضمن تعليمات البرمجة ٨-٢ DTD داخلي لإحصاءات البيسبول. وسوف تقوم بتحويل هذه التعليمات البرمجية لاستخدام DTD خارجي. أولاً قم بتجريد DTD وضعه داخل ملف خاص به. وهذا هو كل شيء بين [<!DOCTYPE SEASON التي تقوم بالفتح و<[للإغلاق. إلا أن <!DOCTYPE SEASON [and]> ليست مضمنة. ويمكن حفظها في ملف اسمه baseball.dtd كما هو موضح في تعليمات البرمجة ٨-١٣. ولا يعتبر الملف هاماً، بالرغم من أن ملحق اسم الملف dtd يعد اصطلاحيا".

تعليمات البرمجة ٨-١٣: ملف DTD للبيسبول

```
<!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
DIVISION)>

  <!- American or National ->
  <!ELEMENT LEAGUE_NAME (#PCDATA)>

  <!- East, West, or Central ->
  <!ELEMENT DIVISION_NAME (#PCDATA)>
  <!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
  <!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
  <!ELEMENT TEAM_CITY (#PCDATA)>
  <!ELEMENT TEAM_NAME (#PCDATA)>
  <!ELEMENT PLAYER (GIVEN_NAME, SURNAME, P, G,
    GS, AB?, R?, H?, D?, T?, HR?, RBI?, SB?, CS?,
    SH?, SF?, E?, BB?, S?, HBP?, CG?, SO?, ERA?, IP?,
HRA?, RA?, ER?, HB?, WP?, B?, WB?, K?)
  >

<!- ====================== ->
  <!- Player Info ->
  <!- Player's last name ->
  <!ELEMENT SURNAME (#PCDATA)>

  <!- Player's first name ->
  <!ELEMENT GIVEN_NAME (#PCDATA)>

  <!- Position ->
  <!ELEMENT P (#PCDATA)>

  <!-Games Played ->
```

```
<!ELEMENT G (#PCDATA)>

<!-Games Started ->
<!ELEMENT GS (#PCDATA)>


<!- ======================== ->
<!- Batting Statistics ->
<!- At Bats ->
<!ELEMENT AB (#PCDATA)>

<!- Runs ->
<!ELEMENT R (#PCDATA)>

<!- Hits ->
<!ELEMENT H (#PCDATA)>

<!- Doubles ->
<!ELEMENT D (#PCDATA)>

<!- Triples ->
<!ELEMENT T (#PCDATA)>

<!- Home Runs ->
<!ELEMENT HR (#PCDATA)>

<!- Runs Batted In ->
<!ELEMENT RBI (#PCDATA)>

<!- Stolen Bases ->
<!ELEMENT SB (#PCDATA)>

<!- Caught Stealing ->
<!ELEMENT CS (#PCDATA)>
```

```
<!- Sacrifice Hits ->
<!ELEMENT SH (#PCDATA)>

<!- Sacrifice Flies ->
<!ELEMENT SF (#PCDATA)>

<!- Errors ->
<!ELEMENT E (#PCDATA)>

<!- Walks (Base on Balls) ->
<!ELEMENT BB (#PCDATA)>

<!- Struck Out ->
<!ELEMENT S (#PCDATA)>

<!- Hit By Pitch ->
<!ELEMENT HBP (#PCDATA)>

<!- ======================= ->
<!- Pitching Statistics ->
<!ر Complete Games ر>
<!ELEMENT CG (#PCDATA)>

<!- Shut Outs ->
<!ELEMENT SO (#PCDATA)>

<!- ERA ->
<!ELEMENT ERA (#PCDATA)>

<!- Innings Pitched ->
<!ELEMENT IP (#PCDATA)>

<!- Home Runs hit Against ->
```

```
<!ELEMENT HRA (#PCDATA)>

<!- Runs hit Against ->
<!ELEMENT RA (#PCDATA)>

<!- Earned Runs ->
<!ELEMENT ER (#PCDATA)>

<!- Hit Batter ->
<!ELEMENT HB (#PCDATA)>

<!- Wild Pitches ->
<!ELEMENT WP (#PCDATA)>

<!- Balk ->
<!ELEMENT B (#PCDATA)>

<!- Walked Batter ->
<!ELEMENT WB (#PCDATA)>

<!- Struck Out Batter ->
<!ELEMENT K (#PCDATA)>

<!- ======================== ->
<!- Fielding Statistics ->
<!- Not yet supported ->
```

وستحتاج فيما يلي إلى تعديل المستند نفسه. فلم يعد إعلان XML مستند مستقل بذاته حيث أنه يعتمد على DTD الموجود في ملف أخر. لذلك لابد أن تتغير سمة Standalone إلى No كما يلي:

```
<?xml version="1.0" standalone="no"?>
```

وعندئذ ينبغي تغيير علامة <!DOCTYPE> حتى يشير إلى DTD بتضمين كلمة SYSTEM الأساسية وURL "عادة ما يكون نسبياً" حيث يتواجد DTD:

```
<!DOCTYPE SEASON SYSTEM "baseball.dtd">
```

وتعتبر بقية المستند هي نفسها السابقة. إلا أن التسجيل الأولي الآن يحتوي فقط على إعـــلان XML وإعلان نوع المستند. ولا يحتوي على DTD. وتوضح تعليمـــات البرمجـــة ٨-١٤ هــذه التعليمات البرمجية.

تعليمات البرمجة ٨-١٤: إحصاءات البيسبول تحتوي على DTD خارجي

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON SYSTEM "baseball.dtd">
<SEASON>
 <YEAR>1998</YEAR>
 <LEAGUE>
  <LEAGUE_NAME>National</LEAGUE_NAME>
  <DIVISION>
   <DIVISION_NAME>East</DIVISION_NAME>
    <TEAM>
     <TEAM_CITY>Atlanta</TEAM_CITY>
     <TEAM_NAME>Braves</TEAM_NAME>
     <PLAYER>
      <GIVEN_NAME>Ozzie</GIVEN_NAME>
      <SURNAME>Guillen</SURNAME>
      <P>Shortstop</P>
      <G>83</G>
      <GS>59</GS>
      <AB>264</AB>
      <R>35</R>
      <H>73</H>
      <D>15</D>
      <T>1</T>
      <HR>1</HR>
      <RBI>22</RBI>
      <SB>1</SB>
      <CS>4</CS>
```

```
      <SH>4</SH>
      <SF>2</SF>
      <E>6</E>
      <BB>24</BB>
      <S>25</S>
      <HBP>1</HBP>
    </PLAYER>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Florida</TEAM_CITY>
    <TEAM_NAME>Marlins</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Montreal</TEAM_CITY>
    <TEAM_NAME>Expos</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>New York</TEAM_CITY>
    <TEAM_NAME>Mets</TEAM_NAME>
  </TEAM>
  <TEAM>
    <TEAM_CITY>Philadelphia</TEAM_CITY>
  <TEAM_NAME>Phillies</TEAM_NAME>
  </TEAM>
</DIVISION>
<DIVISION>
  <DIVISION_NAME>Central</DIVISION_NAME>
  <TEAM>
    <TEAM_CITY>Chicago</TEAM_CITY>
    <TEAM_NAME>Cubs</TEAM_NAME>
  </TEAM>
</DIVISION>
<DIVISION>
  <DIVISION_NAME>West</DIVISION_NAME>
```

```
    <TEAM>
<TEAM_CITY>Arizona</TEAM_CITY>
      <TEAM_NAME>Diamondbacks</TEAM_NAME>
    </TEAM>
  </DIVISION>
 </LEAGUE>
 <LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Baltimore</TEAM_CITY>
      <TEAM_NAME>Orioles</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Chicago</TEAM_CITY>
      <TEAM_NAME>White Sox</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
     <TEAM>
      <TEAM_CITY>Anaheim</TEAM_CITY>
      <TEAM_NAME>Angels</TEAM_NAME>
     </TEAM>
   </DIVISION>
 </LEAGUE>
</SEASON>
```

تأكد من وجود كل من تعليمات البرمجة ٨-١٤ وbaseball.dtd على نفس الدليل ثـــم قــم بتحميل تعليمات البرمجة ٨-١٤ داخل مستعرض الويب كالمعتاد. وإذا تم كل شيء، سترى نفس

الناتج الذي ظهر عند تحميل تعليمات البرمجة ٨-١٢. ويمكنك الآن استخدام DTD نفسه لوصف مستندات أخرى مثل إحصاءات من أعوام أخرى.

وبمجرد إضافة ورقة النمط، سيكون لديك الأجزاء الثلاثة الضرورية للمستند مخزنة داخل ثلاثة ملفات مختلفة. ويتم تطبيق البيانات في ملف المستند والبنية ودلالات الألفاظ على البيانات الموجودة في ملف DTD والتنسيق الموجود في ورقة النمط. وتتيح لك هذه البنية اختيار أو تغيير أي من أو كل هذا بصورة غير تابعة نسبياً.

ويرتبط DTD بالمستند بصورة أقرب من ارتباط المستند بورقة النمط. ويتطلب تغيير DTD بصفة عامة إعادة التحقق من صحة المستند وقد يتطلب تحرير إلى المستند لإعادة مطابقته مع DTD. وتعتمد ضرورة هذا التسلسل على التحرير الذي تقوم به، فإضافة عناصر يعتبر وحدة قضية منفردة، بالرغم من أن إزالة عناصر قد يمثل مشكلة.

تعريفات نوع المستند على URL بعيدة

إذا كان DTD يتم تطبيقه على مستندات متعددة، لن يمكنك دائماً وضع DTD على نفس الدليل مثل كل مستند يستخدم من أجله. وبدلاً من ذلك يمكن استخدام URL لتحديد مكان DTD بدقة. فعلى سبيل المثال، بفرض أن DTD للبيسبول موجود في http://metalab.unc.edu/xml/dtds/baseball.dtd. فيمكنك أن تقوم بالربط به باستخدام علامة <!DOCTYPE> التالية في التسجيل الأولي:

```
<!DOCTYPE SEASON SYSTEM
  "http://metalab.unc.edu/xml/dtds/baseball.dtd">
```

ويستخدم هذا المثال URL كامل وصحيح من أي مكان. وقد ترغب أيضاً في تعيين موضع DTDs المرتبط بالمستند الجذري لخادم الويب أو الدليل الحالي. وبصفة عامة، فإنه يتم قبول أي مرجع يكون URL صحيح ومرتبط بموضع المستند. فعلى سبيل المثال، تغيير إعلانات نوع المستند التالية كلها صحيحة:

```
<!DOCTYPE SEASON SYSTEM "/xml/dtds/baseball.dtd">
<!DOCTYPE SEASON SYSTEM "dtds/baseball.dtd">
<!DOCTYPE SEASON SYSTEM "../baseball.dtd">
```



لا يمكن أن يكون للمستند أكثر من إعلان نوع مستند واحد، أي أكثر من علامة <!DOCTYPE> واحدة ولاستخدام عناصر معلنة في أكثر من DTD واحد، ستحتاج إلى استخدام مراجع كينونات معاملات خارجية. وهو ما سوف يتناوله الفصل التالي.

تعريفات نوع المستند العامة

يقصد بكلمة SYSTEM الأساسية DTD خاصة تستخدم بواسطة شخص واحد أو مجموعة. إلا أن جزء من تعهد XML يتمثل في التلميحات الواسعة التي تعطي مجال كامل مثل ISO أو IEEE يمكنها قياس DTD العامة لتغطية حقولها وهذا التوحيد للمعايير يحمي الأشخاص من وجود مجموعات من العلامات للعناصر أو البنود نفسها ويجعل من السهل على المستخدمين استبدال المستندات القابلة للتشغيل الداخلي.

ولقد تم تصميم DTDs للذين يستخدمون كلمة PUBLIC الأساسية بدلاً من كلمه SYSTEM الأساسية. وكذلك يكون DTD له اسم، كما هو موضح بالعبارة التالية:

```
<!DOCTYPE root_element_name PUBLIC "DTD_name" "DTD_URL">
```

ومرة أخرى، فإن root_element_name هو اسم العنصر الجذري. وPUBLIC هي كلمة XML الأساسية التي تشير إلى DTD هذا للاستخدام على نطاق واسع له اسم. وDTD_name هو الاسم المقترن بهذا DTD. وقد تحاول بعض مستعرضات XML استخدام هذا الاسم لاسترداد DTD من مخزن أو مستودع مركزي. وأخيراً، يعتبر DTD_URL هو URL نسبي أو مطلق حيث يمكن إيجاد DTD إذا لم يمكن استرداده بالاسم من مخزن معروف جيداً.

وتختلف أسماء DTD بعض الشيء عن أسماء XML فقد تحتوي فقط على أحرف ASCII التعدادية الرقمية والمسافة وأحرف إرجاع وأحرف تغذية السطور، والعلامات النقطية التالية: %_$@#*!;?=:/,+() وأيضاً تتبع أسماء DTD العامة بعض العرف.

وإذا كان DTD هو أحد معايير ISO، سيبدأ اسمه بسلسلة "ISO" وإذا اتفق نص المعيار لغير ISO مع DTD، فسيبدأ اسمه بعلامة الجمع (+) وإذا لم يتفق نص المعيار مع DTD، سيبدأ اسمه بواصلة (-). وتكون هذه السلاسل الاستهلالية متبوعة بخط مائل مزدوج (//) واسم صاحب DTD والمتبوع بخط مائل مزدوج أخر ونوع المستند الذي يضعه DTD. وهناك أيضاً خط مائل مزدوج متبوعاً بمعرف لغة ISO 639، مثل EN للإنجليزية. وتتوفر قائمة كاملة بمعرفات ISO 639 من http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt فعلى سبيل المثال، يمكن أن يسمى DTD للبيسبول كما يلي:

```
-//Elliotte Rusty Harold//DTD baseball statistics//EN
```

ويوضح هذا المثال أن DTD ليس نص معيار مصدق عليه (-)، وينتمي إلى Elliotte Rusty Harold، ويصف إحصاءات البيسبول ومكتوب بالإنجليزية. وفيما يلي توضح إعلانات نوع مستند يشير إلى DTD هذا مع هذا الاسم.

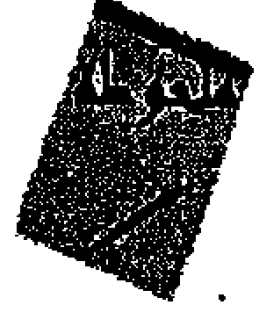
```
<!DOCTYPE SEASON PUBLIC
  "-//Elliotte Rusty Harold//DTD baseball statistics//EN"
```

"http://metalab.unc.edu/xml/dtds/baseball.dtd">

وقد تكون لاحظت أن العديد من محررات HTML مثل BBEdit تقوم آلياً بوضـــع السلســـلة التالية في بداية كل ملف HTML الذي تقوم بإنشائه.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML//EN">

وتوضح هذه السلسلة أن DTD لنص المعيار غير المتفق عليه في XML تم إنتاجه بواســـطة W3C باللغة الإنجليزية.



لا يعتبر W3C من الناحية الفنية تنظيماً للمعايير حيث تتحدد عضويته إلى الشركة التي تقوم بدفع الرسوم الخاصة به، وليس لنصوص مصدق عليها رســـمياً مـــن الجانب الحكومي. وهو يقوم بنشـــر recommendations "أو مـــيزات" وليـــس standards "أو معايير". ولا ترتبط هذه الميزة بالموضوع عملياً.

مجموعات تعريفات نوع المستند الفرعية الداخلية والخارجية

بالرغم من أن أغلب المستندات تتكون من أجزاء تم تعريفها بسهولة فلا تستخدم كل المســـتندات قالباً. وقد تحتاج العديد من المستندات استخدام DTDs قياسية مثل HTML 4.0 DTD في أثنـــاء إضافة عناصر مخصصة لاستخدامها الخاص. وقد تستخدم مستندات أخرى عناصر قياسية فقـط، إلا أنها تحتاج لإعادة ترتيبها. فعلى سبيل المثال، قد تحتوي إحدى صفحات HTML على BODY الذي ينبغي أن يحتوي بالتحديد على رأس H 1 متبوعاً بقائمة تعريف DL بينما قد تحتوي صفحة أخرى على BODY يحتوي على رؤوس عديدة مختلفة وفقرات وصور بترتيب غير محـــدد. وإذا كان لمستند بعينه بنية مختلفة عن الصفحات الأخرى الموجودة على الموقع، فقد يكون من المفيـــد تعريف بنيته في المستند نفسه وليس في DTD منفصل. وتجعل هذه الطريقة تحرير المستند أكـــثر سهولة.

وقد يستخدم المستند كل من DTD داخلي وخارجي وتكون الإعلانات الداخلية داخل أقـــواس مربعة بنهاية علامة <!DOCTYPE>. فعلى سبيل المثال، بفرض أنك تريـــد صفحـــة تتضمـــن إحصاءات البيسبول وتحتوي أيضاً على رأس وتذييل. وقد يبدو مثل هذا المستند كما هو موضـــح في تعليمات البرمجة ٨-١٥. وقد تـــم تجميـــع المعلومـــات الخاصـــة بالبيســـبول مـــن ملـــف baseball.dtd الذي يشكل مجموعة DTD الفرعية الخارجية. ويأتي تعريف العنصر الجـــذري DOCUMENT وعنصري TITLE وSIGNATURE من مجموعـــة DTD الفرعـــية الداخليـــة المضمنة في المستند نفسه. وقد يعتبر ذلك شيئاً غير مألوف يعطي الشيء والأكثر شيوعاً الأجزاء العامة من المحتمل أن تكون جزء من DTD خارجي بينما تكـــون الأجـــزاء الداخليـــة محـــددة الموضوع بصورة أكثر وضوحاً.

تعليمات البرمجة ٨-١٥ : مستند بالبيسبول يحتوي DTD الخاص به على مجموعتين فرعيتين داخلية وخارجية

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT SYSTEM "baseball.dtd" [
  <!ELEMENT DOCUMENT (TITLE, SEASON, SIGNATURE)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT COPYRIGHT (#PCDATA)>
  <!ELEMENT EMAIL (#PCDATA)>
  <!ELEMENT LAST_MODIFIED (#PCDATA)>
  <!ELEMENT SIGNATURE (COPYRIGHT, EMAIL, LAST_MODIFIED)>
]>
<DOCUMENT>
 <TITLE>1998 Major League Baseball Statistics</TITLE>
 <SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
   <LEAGUE_NAME>National</LEAGUE_NAME>
   <DIVISION>
     <DIVISION_NAME>East</DIVISION_NAME>
<TEAM>
       <TEAM_CITY>Atlanta</TEAM_CITY>
       <TEAM_NAME>Braves</TEAM_NAME>
      </TEAM>
      <TEAM>
       <TEAM_CITY>Florida</TEAM_CITY>
       <TEAM_NAME>Marlins</TEAM_NAME>
      </TEAM>
      <TEAM>
       <TEAM_CITY>Montreal</TEAM_CITY>
       <TEAM_NAME>Expos</TEAM_NAME>
      </TEAM>
      <TEAM>
```

```
        <TEAM_CITY>New York</TEAM_CITY>
        <TEAM_NAME>Mets</TEAM_NAME>
      </TEAM>
      <TEAM>
        <TEAM_CITY>Philadelphia</TEAM_CITY>
      <TEAM_NAME>Phillies</TEAM_NAME>
      </TEAM>
  </DIVISION>
  <DIVISION>
     <DIVISION_NAME>Central</DIVISION_NAME>
      <TEAM>
        <TEAM_CITY>Chicago</TEAM_CITY>
        <TEAM_NAME>Cubs</TEAM_NAME>
      </TEAM>
  </DIVISION>
  <DIVISION>
     <DIVISION_NAME>West</DIVISION_NAME>
      <TEAM>
        <TEAM_CITY>Arizona</TEAM_CITY>
        <TEAM_NAME>Diamondbacks</TEAM_NAME>
      </TEAM>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
     <DIVISION_NAME>East</DIVISION_NAME>
      <TEAM>
        <TEAM_CITY>Baltimore</TEAM_CITY>
        <TEAM_NAME>Orioles</TEAM_NAME>
      </TEAM>
  </DIVISION>
  <DIVISION>
     <DIVISION_NAME>Central</DIVISION_NAME>
```

```
      <TEAM>
        <TEAM_CITY>Chicago</TEAM_CITY>
        <TEAM_NAME>White Sox</TEAM_NAME>
      </TEAM>
    </DIVISION>
    <DIVISION>
      <DIVISION_NAME>West</DIVISION_NAME>
      <TEAM>
        <TEAM_CITY>Anaheim</TEAM_CITY>
        <TEAM_NAME>Angels</TEAM_NAME>
      </TEAM>
    </DIVISION>
  </LEAGUE>
</SEASON>
<SIGNATURE>
  <COPYRIGHT>Copyright 1999 Elliotte Rusty Harold</COPYRIGHT>
  <EMAIL>elharo@metalab.unc.edu</EMAIL>
  <LAST_MODIFIED>March 10, 1999</LAST_MODIFIED>
</SIGNATURE>
</DOCUMENT>
```

وعند تعارض العناصر التي تحمل نفس الاسم في مجموعات DTD الفرعية الداخلية والخارجية، تكون الأسبقية للعناصر التي تم إعلانها داخلياً. وتوفر هذه الأسبقية تقنية مشابهة جزئية. فعلى سبيل المثال، بفرض أنك تريد تجاوز تعريف عنصر PLAYER حتى يمكن أن يحتوي فقط على إحصاءات الرمي. ويمكنك استخدام أغلب الإعلانات نفسها في DTD كالبيسبول، مع تغيير عنصر PLAYER كما يلي:

```
<!DOCTYPE SEASON SYSTEM "baseball.dtd" [
  <!ELEMENT PLAYER (GIVEN_NAME, SURNAME, P, G,
    GS, AB?, R?, H?, D?, T?, HR?, RBI?, SB?, CS?,
    SH?, SF?, E?, BB?, S?, HBP?)
  >
]>
```

خلاصة

لقد تعرفت في هذا الفصل على كيفية استخدام DTD لوصف بنية مستند ، أي كل من العناصر المطلوبة والخيارية التي يحتويها وكيفية ارتباط هذه العناصر بمستند أخر وبالتحديد، لقد تناولنا في هذا الفصل ما يلي:

- ♦ يزودك تعريف نوع المستند DTD بقائمة بالعناصر والعلامات والسمات والمكونات التي يحتويها المستند والعلاقة بينها وبين أخرى.
- ♦ قد يحتوي التسجيل الأولى للمستند على إعلان نوع المستند الذي يحدد العنصر الجذري ويحتوي على DTD. وهو يوضع بين إعلان XML وقبل بداية المستند الفعلي. ويتم تحديده بواسطة ,<!DOC-TYPE ROOT [and]> هو اسم العنصر الجذري.
- ♦ تقوم DTDs بتخطيط العلامات المسموح بها والبينة للمستند. ويعتبر المستند، الذي يلتزم بقواعد DTD الخاص به، مستنداً صحيحاً.
- ♦ توضح إعلانات نوع المستند اسم وتوابع العنصر.
- ♦ لابد أن تظهر التوابع المنفصلة بفواصل والموجودة في إعلان نوع المستند، بنفس الترتيب في هذا العنصر داخل المستند.
- ♦ تعني علامة الجمع احتمال ظهور مثال واحد أو أكثر للعنصر.
- ♦ تعني العلامة النجمية احتمال عدم ظهور أمثلة أو ظهور أكثر من مثال للعنصر.
- ♦ تعني علامة الاستفهام احتمال عدم ظهور أمثلة أو ظهور أكثر من مثال للتابع.
- ♦ يعني الشريط العمودي وجوب استخدام عنصر واحد أو آخر.
- ♦ تجمع الأقواس العناصر التابعة وذلك لتسمح بإعلانات عناصر أكثر تفصيلاً.
- ♦ يحتوي المحتوى المختلط على كل من العناصر وبيانات الأحرف الموزعة إلا أنه يحدد البنية التي يمكن تشكيلها في العنصر الأصل.
- ♦ يتم إعلان العناصر الفارغة بواسطة كلمة EMPTY الأساسية.
- ♦ التعليقات تجعل DTDs أكثر وضوحاً.
- ♦ يمكن تعيين موضع DTDs الخارجية لمستخدم كلمة SYSTEM الأساسية وURL في إعلان نوع المستند.
- ♦ يمكن تعيين موضع DTDs القياسية باستخدام كلمة PUBLIC الأساسية في إعلان نوع المستند.
- ♦ تتجاوز الإعلانات، داخل مجموعة DTD الفرعية الداخلية، الإعلانات المتعارضة في مجموعة DTD الفرعية الخارجية.

وسوف تتعرف في الفصل التالي على المزيد عن DTDs، بما في ذلك كيفية توفـــير مراجــع الكينونات للنص الاستبدالي وكيفية فصل DTDs عن المستندات التي تقوم بوصفها حتى يكون من السهل إشراكها بين المستندات وستتعرف أيضاً على كيفية استخدام DTDs متعددة لوصف مسـتند واحد.

٩

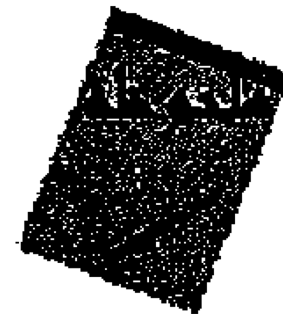
المجموعات الفرعية للكينونات وتعريفات نوع المستند الخارجية

قد يتلقى أحد مستندات HML كل من البيانات والإعلانات من عدة مصادر مختلفة في ملفات متعددة. وفي الواقع قد يتم تلقي بعض البيانات مباشرة من قواعد البيانات، أو نصوص CGI أو أية مصادر أخرى غير ملفية. ويطلق على العناصر التي يتم تخزين الأجزاء الخاصة بأحد ملفات HML داخل أي نموذج، يطلق عليها كينونات. وتقوم مراجع الكينونات بتحميل هذه الكينونات داخل مستند XML الرئيسي. وتقوم مراجع الكينونات العامة بتحميل بيانات داخل العنصر الجذري من أحد مستندات XML، بينما تقوم مراجع كينونات المعاملات بتحميل البيانات داخل تعريفات نوع المستند الخاصة بالمستند (DTD).

ما هي الكينونات

من المنطقي أن يتم إعداد أحد مستندات XML من تسجيل تقديمي متبوعا بعنصر جذري يحتوي بصورة تامة على كل العناصر الأخرى. إلا أننا إذا تحدثنا بطريقة عملية، فليس من الممكن أن تنفصل البيانات الفعلية لأحد مستندات XML في عدة ملفات. فعلى سبيل المثال، قد يظهر كل عنصر من عناصر PLAYER في ملف منفصل، حتى على الرغم من احتواء العنصر الجذري على اللاعبين التسعمائة في إحدى المجموعات. ويطلق على وحدات التخزين التي تحتوي على أجزاء معينة من أحد مستندات XML، الكينونات. وقد يتكون أحد هذه الكينونات من ملف أو سجل قاعدة بيانات أو أي عنصر آخر يحتوي على بيانات. فعلى سبيل المثال، تعتبر كل ملفات XML الكاملة الموجودة في هذا الكتاب كينونات.

ويطلق مصطلح كينونة المستند document entity على وحدة التخزين التي تحتوي على إعلان XML وإعلان نوع المستند والعنصر الجذري. إلا أن العنصر الجذري وما يندرج أسفله قد يحتوي على مراجع للكينونات تشير إلى بيانات إضافية، لابد من إدراجها داخل المستند. ويقوم معالج XML للتحقق من الصحة بتجميع كل الكينونات ذات المراجع المختلفة داخل ملف منطقي واحد قبل أن يقوم بنقل المستند إلى التطبيق النهائي أو عرض الملف.



قد يحدث وليس من الضروري، أن تقوم المعالجات التي لا تتحقق من صحة، بإدراج كينونات خارجية. إلا أنها لابد أن تقوم بإدراج الكينونات الداخلية.

وتعتبر غاية الكينونات الأساسية هي وجود: XML صحيح التكوين أو النماذج الأخرى للنصوص أو البيانات الثنائية. ويعد كل من التسجيل الأولي وإعلان نوع المستند جزء من العنصر الجذري للمستند الذي ينتميان إليه. وتعتبر ورقة نمط XML أحد العناصر وذلك لأنها فقط أحد مستندات XML صحيحة التكوين. ولا يعتبر العنصر الذي يقوم بتنظيم ورقة النمط أحد العناصر التي تقوم بإعداد مستند XML الذي يتم تطبيق ورقة النمط عليه.

ولأغلب العناصر أسماء يمكنك الإشارة إلى العناصر بواسطتها والاستثناء الوحيد هو كينونـــة المستند، وهي الملف الرئيسي الذي يحتوي على ملف XML "بالرغم من أنه ليست هناك حاجــــة إلى أن يكون ذلك ملف مقابل لسجل بيانات أو ناتج برامج CGI أو أي شـــيء أخــر". وكينونـــة المستند هي وحدة التخزين التي تأخذ أي شكل وتحتوي على تعريف XML، وتعريف نوع المستند "إن وجد"، والعنصر الجذري. ولذلك فإن كل مستند XML يكون له على الأقل كينونـــة واحـــدة. والكينونات نوعان، إما داخلية أو خارجية. ويتم تعريف الكينونات الداخلية بالكامل داخل كينونـــة المستند. ويعتبر المستند نفسه هو أحد هذه الكينونات، لذلك فإن كل مستند XML يكون لـــه علـــى الأقل كينونة واحدة داخلية.

وتأخذ الكينونات الخارجية مضمونها، على العكس من ذلك، من مصدر أخــــر يتـم تحديـد موضعه بواسطة URL. ويشتمل المستند الرئيسي فقط على مرجع إلـــى URL، حيـث يتواجـد المضمون الفعلي. وفي HTML يقوم عنصر IMG بتقديم كينونة خارجية "بيانات الصورة الفعلية"، بينما يعتبر المستند نفسه الذي يتم تضمينه بين علامتي <HTML> و</HTML> كينونة داخلية.

وتنقسم الكينونات إلى فئتين: موزعة وغير موزعة. وتحتوي الكينونات الموزعة على نـــص XML صحيح التكوين. وأما الكينونات غير الموزعة فتحتوي إما على بيانات ثنائية أو نص ليـس XML "مثل رسائل البريد الإلكتروني" ولا يتوفر في الوقت الحالي دعماً قويـــا للكينونـــات غـــير الموزعة من معظم معالجات XML. وسوف نركز أكثر في هذا الفصل على تنـــاول الكينونـــات الموزعة.



يتناول الفصل ١١ "إضافة بيانات ليست من نوع XML" الكينونات غير الموزعة.

وتتيح مراجع الكينونات للبيانات من كينونات متعددة أن يتم دمجها ببعضها لتكويـــن مســـتند واحد. وتقوم مراجع الكينونات العامة بدمج البيانات داخل محتوى المستند. وتقوم مراجع كينونـــة المعامل بدمج التعريفات داخل DTD الخاص بالمستند. وتعتـــبر &quote; و& و&pos; و> و< مراجع كينونات تم تعريفها مسبقا تشير إلى كينونات النص > و< و' و" و& على التوالي. إلا أنه يمكن أيضا تعريف كينونات جديدة في DTD الخاص بمستندك.

الكينونات الداخلية العامة

يمكن اعتبار مرجع الكينونة العامة الخارجية، اختصارا لنص يستخدم بصورة شـــائعة، أو نـــص يصعب طباعته. وتقوم علامة <!ENTITY> في DTD بتعريف الاختصار والنص الذي يشـــير

إليه الاختصار. فعلى سبيل المثال، بدلاً من طباعة نفس التذييل بأسفل كل صفحة، يمكنك ببساطة تعريف النص بكينونة Footer بأسفل كل صفحة ثم طباعة &Footer; بأسفل كل صفحة. وأيضا إذا قررت تغيير قالب التذييل "مثلا بسبب تغيير عنوان بريدك الإلكتروني"، فستحتاج لإجراء التغيير مرة واحدة فقط في DTD بدلاً من كل صفحة تحتوي عل التذييل.

وتبدأ مراجع الكينونات العامة بعلامة الضم (&) وتنتهي بفاصلة منقوطة (;) مع اسم الكينونة بين هذين الحرفين. فعلى سبيل المثال، تعتبر < مرجع كينونة عام لعلامة أقل من (<)، واسم هذه الكينونة هو Lt. والنص الاستبدالي لهذه الكينونة هو سلسلة الحرف الواحد <. وتتكون أسماء الكينونات من مجموعة من الحروف الهجائية العددية وتسطير أسفل السطر. وأما المسافات البيضاء والحروف النقطية فهي ممنوعة. ومثل أغلب الأشياء الأخرى في XML، تعتبر مراجع الكينونات تحسين لحالة الأحرف.



بالرغم من أن علامة النقطتين (:) متاحة فنياً في أسماء الكينونات، فهناك تحفظ حول استخدامها مع مسافات الأسماء، وهو ما نوضحه في الفصل ١٨.

تعريف مراجع الكينونات الداخلية العامة

يتم تعريف مراجع الكينونات الداخلية العامة في DTD بعلامة <!ENTITY> والتي لها التنسيق التالي:

```
<!ENTITY name "replacement text">
```

ويعتبر *name* هو اختصار *replacement text*. وينبغي إحاطة النص الاستبدالي بعلامتي تنصيص حيث أنها قد تحتوي على مسافات بيضاء وترميز XML. فتقوم أنت بطباعة اسم الكينونة في المستند إلا أن القارئ يرى النص الاستبدالي. فعلى سبيل المثال، بفرض أن شخص يحمل اسم "Elliotte Rusty Harold"، فقد يظل يطبع هذه العبارة. إلا أنه يمكن تعريف مرجع كينونة عام بالاسم حتى يرى القارئ عبارة "Elliotte Rusty Harold" عند طباعة &ERH; في أي وقت. وهذا التعريف هو:

```
<!ENTITY ERH "Elliotte Rusty Harold">
```

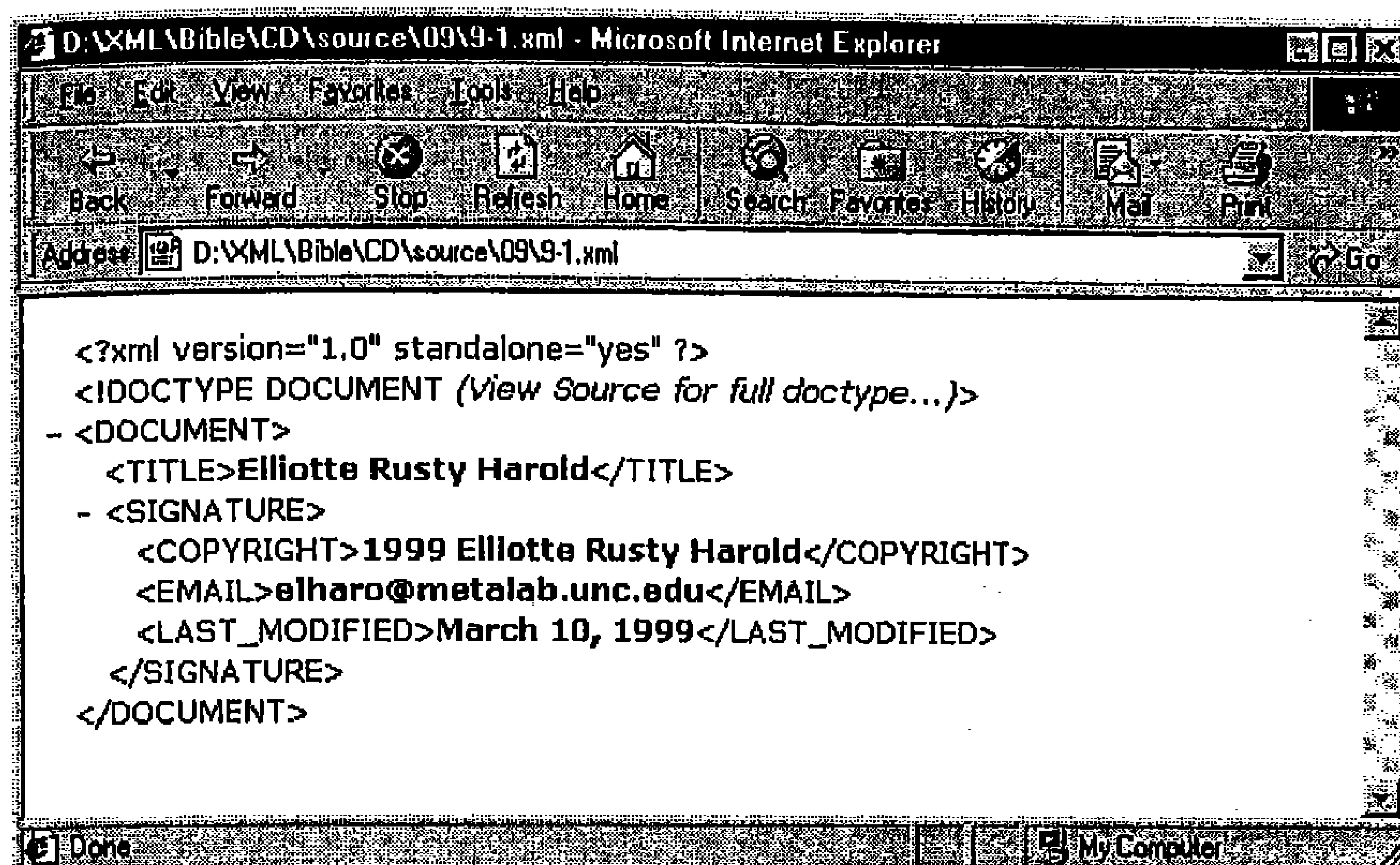
وتوضح تعليمات البرمجة ٩-١ مرجع الكينونة العام &ERH; بينما يوضح الشكل ٩-١ هذا المستند وقد تم تحميله داخل Internet Explorer. وسوف ترى أن مرجع الكينونة &ERH; في تعليمات البرمجة المصدر يتم استبدالها بعبارة Elliotte Rusty Harold في الناتج.

تعليمات البرمجة ٩-١: مرجع الكينونة العامة الداخلية ERH

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [

  <!ENTITY ERH "Elliotte Rusty Harold">

  <!ELEMENT DOCUMENT (TITLE, SIGNATURE)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT COPYRIGHT (#PCDATA)>
  <!ELEMENT EMAIL (#PCDATA)>
  <!ELEMENT LAST_MODIFIED (#PCDATA)>
  <!ELEMENT SIGNATURE (COPYRIGHT, EMAIL, LAST_MODIFIED)>
]>
<DOCUMENT>
  <TITLE>&ERH;</TITLE>
  <SIGNATURE>
    <COPYRIGHT>1999 &ERH;</COPYRIGHT>
    <EMAIL>elharo@metalab.unc.edu</EMAIL>
    <LAST_MODIFIED>March 10, 1999</LAST_MODIFIED>
  </SIGNATURE>
</DOCUMENT>
```



الشكل ٩-١ تعليمات البرمجة ٩-١ بعد استبدال مرجع الكينونة الداخلية العامة بالكينونة الفعلية.

ولاحظ أن مرجع الكينونة العامة، &ERH; يظهر داخل كل من عنصري COPYRIGHT وTITLE حتى على الرغم من أنه قد تم إعلان الاثنين ليقبلا فقط #PCDATA كتوابع. ويعتبر هذا الترتيب صحيح حيث أن النص الاستبدالي لمرجع الكينونة &ERH; هو بيانات أحرف موزعة. ويتم إجراء التحقق من الصحة للمستند بعد استبدال جميع مراجع الكينونات بالقيم الخاصة بها.

ويحدث نفس الشيء عند استخدام ورقة نمط. فالأنماط يتم تطبيقها على شجرة العناصر عند تواجدها بعد استبدال مراجع الكينونات بقيم الكينونات.

ويمكنك اتباع نفس الوضع للإعلان عن مراجع الكينونات العامة لحقوق النشر وعنوان البريد الإلكتروني وتاريخ أخر تعديل:

```
<!ENTITY COPY99 "Copyright 1999">
<!ENTITY EMAIL "elharo@metalab.unc.edu">
<!ENTITY LM "Last modified: ">
```

ولقد تجاهلنا التاريخ في كينونة &LM; حيث من المحتمل أن تتغير من مستند لأخر ولا توجد أية مميزات لجعلها مرجع كينونة.

ويمكنك الآن إعادة كتابة جزء المستند من تعليمات البرمجة ٩-١ بطريقة أكثر إحكاماً:

```
<DOCUMENT>
  <TITLE>&ERH;</TITLE>
  <SIGNATURE>
   <COPYRIGHT>&COPY99; &ERH;</COPYRIGHT>
   <EMAIL>&EMAIL;</EMAIL>
   <LAST_MODIFIED>&LM; March 10, 1999</LAST_MODIFIED>
  </SIGNATURE>
</DOCUMENT>
```

ومن مميزات استخدام مراجع الكينونات، بدلاً من النص الكامل، أن هذه المراجع تجعل تغيير النص أكثر سهولة. وتتمثل هذه الفائدة على وجه الخصوص عند مشاركة DTD واحد بين مستندات متعددة. فعلى سبيل المثال، بفرض أنك تريد استخدام عنوان البريد الإلكتروني eharold@solar.stanford.edu بدلاً من eharold@solar.stanford.edu. فبدلاً من البحث والاستبدال عبر ملفات متعددة، يمكنك ببساطة تغيير سطر واحد من DTD كما يلي:

```
<!ENTITY EMAIL "eharold@solar.stanford.edu">
```

استخدام مراجع الكينونات العامة في تعريف نوع المستند

قد تتساءل عما إذا كان من الممكن تضمين مرجع كينونة عامة واحد داخل مرجع آخر كما يلي:

```
<!ENTITY COPY99 "Copyright 1999 &ERH;">
```

وفي الواقع، فإن هذا المثال صحيح، حيث أن كينونة ERH تظهر جزء من كينونة COPY99 والتي ستصبح فيما بعد هي نفسها جزءاً من محتوى المستند. ويمكن أيضاً استخدام مراجع الكينونات العامة في مواضع أخرى في DTD والتي تصبح في النهاية جزء من محتوى المستند "مثل قيمة سمة افتراضية"، وذلك بالرغم من وجود قيود. وأول قيد هو أن العبارة لا يمكن أن تستخدم مرجعاً معاداً، مثل المرجع التالي:

```
<!ENTITY ERH "&COPY99 Elliotte Rusty Harold">
<!ENTITY COPY99 "Copyright 1999 &ERH;">
```

والقيد الثاني هو: أن مراجع الكينونات العامة قد لا تقوم بإدراج نص يعتبر فقط جزء من DTD ولا يتم استخدامه كجزء من محتوى المستند. فعلى سبيل المثال، لم يصلح اختصار المحاولة التالية:

```
<!ENTITY  PCD    "(#PCDATA)">
<!ELEMENT ANIMAL &PCD;>
<!ELEMENT FOOD   &PCD;>
```

إلا أنه غالباً ما يكون من الأفضل جعل مراجع الكينونات تقوم بدمج النص داخل DTD الخاص بالمستند. ولهذا الغرض، يستخدم XML مرجع كينونة المعامل والموضحة لاحقاً في هذا الفصل.

والقيد الوحيد على قيم الكينونات العامة هو: أنها قد لا تحتوي على الثلاثة أحرف % و& و" مباشرة بالرغم من إمكانية تضمينها عن طريق مراجع الأحرف. وقد يتم تضمين & و% إذا كانا يبدءان مرجع كينونة بدلاً من تقديم نفسها ببساطة. ويعني نقص القيود المفروضة احتمال احتواء الكينونة على علامات وشغلها لأسطر متعددة. فعلى سبيل المثال، تعتبر كينونة SIGNATUR التالية صحيحة:

```
<!ENTITY SIGNATURE
"<SIGNATURE>
   <COPYRIGHT>1999 Elliotte Rusty Harold</COPYRIGHT>
   <EMAIL>elharo@metalab.unc.edu</EMAIL>
   <LAST_MODIFIED>March 10, 1999</LAST_MODIFIED>
  </SIGNATURE>"
>
```

وسؤالنا الآن هو ما إذا كان من الممكن احتواء الكينونات على معاملات. فهل يمكن استخدام كينونة SIGNATUR الموجودة بأعلى وتغيير البيانات في كل عنصر منفصل من LAST-Modified بكل صفحة؟ والإجابة على هذا التساؤل هي قطعاً لا، حيث أن الكينونات تكون فقط للنص الاستبدالي الثابت. وإذا كنت تريد نقل بيانات إلى إحدى الكينونات، فينبغي استخدام علامة مع إرشادات الأداء المناسبة في ورقة النمط.

مراجع الكينونات العامة المعرفة مسبقاً

يقوم XML بتعريف خمسة مراجع كينونات عامة كما هو موضح في الجدول ٩-١ وتظهر مراجع الكينونات الخمسة هذه في مستندات XML في موضع أحرف معينة التي يتم ترجمتها كترميز. فعلى سبيل المثال، يشير مرجع الكينونة < إلى علامة الأقل من (<)، التي يمكن ترجمتها كبداية إحدى العلامات.

وللحصول على أعلى مستوى للتوافق يجب الإعلان عن هذه المراجع في DTD إذا كنت تنوي استخدامها. وتعتبر الإعلانات دقيقة جدا بالفعل، حيث يجب تخطي الأحرف في DTD دون تكرار. وللإعلان عن هذه المراجع، استخدم تعليمات البرمجة ٩-٢ التي توضح الإعلانات الهامة:

الجدول ٩-١

مراجع كينونات XML المعرفة مسبقاً

الحرف	مرجع الكينونة
&	&
<	<
>	>
"	"
'	'

تعليمات البرمجة ٩-٢ إعلانات خاصة بمراجع الكينونات العامة المعرفة مسبقاً

```
<!ENTITY lt     "&#38;#60;">
<!ENTITY gt     "&#62;">
<!ENTITY amp    "&#38;#38;">
<!ENTITY apos   "&#39;">
<!ENTITY quot   "&#34;">
```

الكينونات الخارجية العامة

تعتبر الكينونات الخارجية هي بيانات موجودة خارج الملف الرئيسي الذي يحتوي على العنصر الجذري/كينونة المستند. وتتيح مراجع الكينونات الخارجية تضمين هذه الكينونات الخارجية في مستندك وإنشاء مستندات XML من ملفات تابعة متعددة.

وتشبه المستندات التي تستخدم بها الكينونات الداخلية فقط وضع HTML بصورة واضحة. يتوفر النص الكامل للمستند في ملف فردي. وقد يتم ربط الصور والتطبيقات الصغيرة والأصوات والبيانات الأخرى التي ليست بيانات HTML، ولكن على الأقل يتواجد النص كله. وهناك بالطبع بعض المشكلات المرتبطة بوضع HTML. فعلى وجه التحديد، يكون من الصعب تضمين المعلومات الحيوية باستخدام CGI وتطبيقات Java الصغيرة وبرنامج قواعد البيانات المتطورة واحتواءات جانب الخادم ووسائل أخرى متنوعة، إلا أن HTML وحده الذي يقدم مستند

ثابت. وينبغي عليك الانتقال خارج HTML لإنشاء مستند من أجزاء متعددة. وقد تكون الأطر هي أبسط حلول HTML لهذه المشكلة، إلا أنها تعتبر مشكلة كبيرة دائماً ما تسبب للمستخدمين الارتباك والقلق.

ويتمثل جزء من المشكلة في عدم احتواء أحد ملفات HTML بصورة طبيعية داخل ملف أخر. فينبغي أن يكون لكل مستند HTML بالتحديد BODY واحد وليس أكثر. وتتيح محتويات جانب الخادم فقط تضمين أجزاء من HTML ولا تتيح أبدا تضمين مستند كامل صحيح داخل المستند. وبالإضافة إلى ذلك يعتبر ما يتضمنه جانب الخادم هو تابع للخادم وليست جزءاً من HTML بالفعل.

إلا أن XML يعتبر أكثر مرونة. فليس بالضرورة أن يكون العنصر الجذري لمستند هو نفسه العنصر الجذري لمستند أخر. وحتى لو كان كلا المستندان يشتركان في نفس العنصر الجذري، فقد يعلن DTD أن العنصر يتاح احتواؤه لنفسه. ولا يمنع معيار XML مستندات XML صحيحة التكوين من أن يتم تضمينها بمستندات XML صحيحة التكوين أخرى عندما يتاح ذلك.

إلا أن XML يذهب إلى ما هو أبعد، فبتعريف التقنيات التي بها يمكن إنشاء مستندات XML أصغر موجودة إما على نظام محلي أو بعيد. ويعتبر الموزع مسؤولاً عن دمج كل المستندات المختلفة معاً بترتيب مختلط. وقد تحتوي المستندات على مستندات أخرى والتي بدورها قد تحتوي على مستندات أخرى. وطالما لا يوجد رفض "خطأ تم تقريره بواسطة المعالج"، فسيرى التطبيق مستند واحد متكامل. وفي الأساس يوفر ذلك ما يتضمنه جانب العميل.

وباستخدام XML يمكنك استخدام مرجع كينوية خارجية لتضمين مستند واحد بآخر. وفي DTD ستقوم بإعلان المرجع الخارجي باستخدام العبارة التالية:

```
<!ENTITY name SYSTEM "URI">
```



المقصود باختصار (URI) Uniform Resource Identifier وتعتبر URIs شبيهة URLs، إلا أنها تتيح مواصفات أكثر إحكاماً للمصادر المرتبطة. ونظرياً، تفصل URLs المصدر عن الموقع حتى يمكن لمستعرض الويب تحديد ارتباط واضح لهذا العاكس. وتعتبر URLs هي منطقة للبحث النشط والمناقشات أو المناظرات الحادة. لذلك فإن كل من URIs وURLs الموجودة في هذا الكتاب تصلح عملياً لكل الأغراض.

فعلى سبيل المثال، قد تريد وضع قالب الإرشادات بكل صفحة من الموقع تقريباً. فمن أجل عدم التعريف على نحو معين، لنفترض أن قالب الإرشادات هو تعليمات البرمجية الموضحة في تعليمات البرمجة ٩-٣. وأيضا لنفترض أنه يمكنك استرداد هذه التعليمات البرمجية من URL.

http://metalab.unc.edu/xml/signature.xml.

تعليمات البرمجة ٩-٣: ملف إرشادات XML

```
<?xml version="1.0"?>
<SIGNATURE>
  <COPYRIGHT>1999 Elliotte Rusty Harold</COPYRIGHT>
  <EMAIL>elharo@metalab.unc.edu</EMAIL>
</SIGNATURE>
```

ويمكن إقران هذا الملف بمرجع الكينونة ;&SIG بإضافة الإعلان التالي إلى DTD:

```
<!ENTITY SIG SYSTEM "http://metalab.unc.edu/xml/signature.xml">
```

ويمكن أيضا استخدام URL نسبي. فعلى سبيل المثال:

```
<!ENTITY SIG SYSTEM "/xml/signature.xml">
```

وإذا كان الملف المراد تضمينه موجود على نفس الدليل الموجود عليه الملف الذي يتضمنه فستحتاج فقط لاستخدام اسم الملف. فعلى سبيل المثال:

```
<!ENTITY SIG SYSTEM "signature.xml">
```

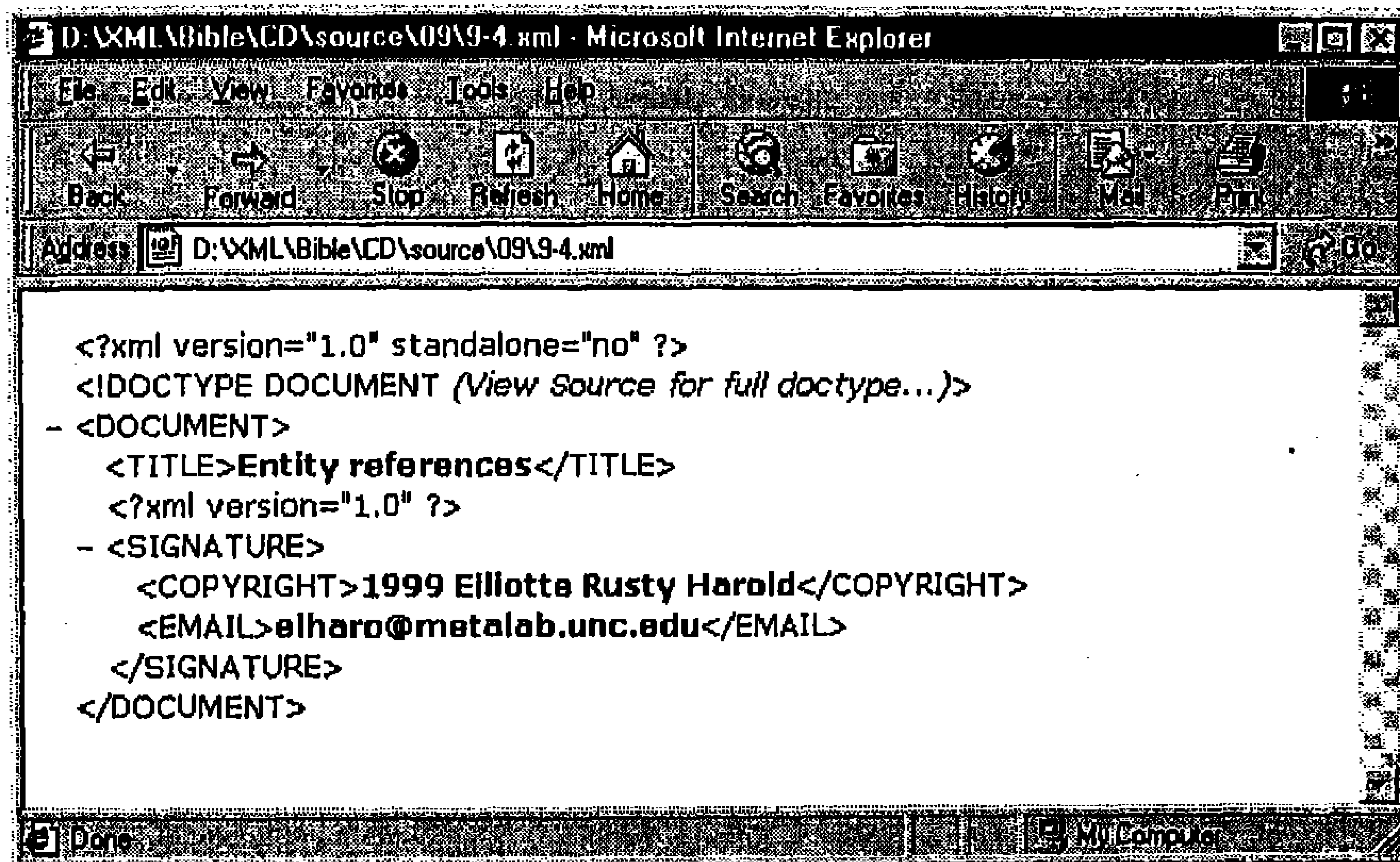
وباستخدام أي من هذه الإعلانات يمكن تضمين محتويات ملف الإرشادات بمستند في أي نقطة فقط باستخدام ;&sig كما هو موضح مع المستند البسيط في تعليمات البرمجة ٩-٤. ويوضح الشكل ٩-٢ المستند الناتج في Internet Explorer 5.0.

تعليمات برمجة ٩-٤: مرجع الكينونة الخارجية العامة SIG

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT (TITLE, SIGNATURE)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT COPYRIGHT (#PCDATA)>
  <!ELEMENT EMAIL (#PCDATA)>
  <!ELEMENT SIGNATURE (COPYRIGHT, EMAIL)>
```

```
  <!ENTITY SIG SYSTEM
    "http://metalab.unc.edu/xml/signature.xml">
]>
<DOCUMENT>
  <TITLE>Entity references</TITLE>
  &SIG;
  </DOCUMENT>
```

وبجانب إضافة مرجع الكينونة الخارجية، لاحظ أن سمة Standalone لإعـــلان XML لـــها الآن قيمة no حيث أن هذا الملف لم يعد كاملا. ويتطلب توزيع أو تقسيم الملف بيانات إضافيـــة من ملف .Signature.xml الخارجي.



الشكل ٩-٢ مستند يستخدم مرجع كينونة خارجية عامة.

كينونات المعاملات الداخلية

أصبحت الكينونات العامة جزءاً من المستند وليس DTD. ويمكن استخدامها داخل DTD ولكـــن فقط في المواضع التي ستصبح فيها جزءاً من نص المستند. وقد لا تقوم مراجع الكينونات العامـــة بإدراج نص يعتبر فقط جزء من DTD ولا يستخدم كجزء من محتوى المستند. إلا أنه من المفيـــد في بعض الأحيان وجود مراجع كينونات داخل DTD. ولهذا الغرض يوفر XML ما يسمى بـــــ *parameter entity reference* أو مرجع كينونة المعامل.

وتتشابه مراجع كينونات المعاملات مع مراجع الكينونات العامة مع وجود الاختلافين التاليين:

١- تبدأ مراجع كينونات المعاملات بالعلامة المئوية (%) بدلاً من علامة الضم (&).

٢- يمكن أن تظهر مراجع كينونات المعاملات فقط في DTD وليس في محتوى المستند.

ويتم إعلان كينونات المعاملات داخل DTD مثل الكينونات العامة مع إضافة العلامة المئوية قبل الاسم. وتبدو العبارة كما يلي:

```
<!ENTITY % name "replacement text">
```

ويعتبر الاسم اختصاراً للكينونة. ويستخدم القارئ النص الاستبدالي الذي يجب ظهوره داخل علامات تنصيص. فعلى سبيل المثال:

```
<!ENTITY % ERH "Elliotte Rusty Harold">
<!ENTITY COPY99 "Copyright 1999 %ERH;">
```

وتعمل محاولتنا السابقة لاختصار (#PCDATA) عندما يتم استبدال مرجع الكينونة العام بمرجع كينونة المعامل:

```
<!ENTITY % PCD "(#PCDATA)">
<!ELEMENT ANIMAL %PCD;>
<!ELEMENT FOOD %PCD;>
```

وتظهر القيمة الحقيقية لمراجع كينونات المعاملات في القوائم الشائعة المشتركة بالتوابع والسمات بين العناصر. وكلما زاد حجم قالب النص الذي تستبدله وكلما زاد عدد مرات استخدامه، زادت فائدة مراجع كينونات المعاملات. فعلى سبيل المثال، بفرض أن DTD يعلن عدد من عناصر حاوية مستوى القالب مثل PARAGRAPH وHEADING وCell. فقد يحتوي كل من عناصر الحاوية هذه على رقم معرف بالعناصر المحمولة مثل PERSON وDEGREE وMODEL, PRODUCT, ANIMAL, INGREDIENT، وهكذا. وقد تظهر إعلانات العناصر لعناصر الحاوية كما يلي:

```
<!ELEMENT PARAGRAPH
  (PERSON | DEGREE | MODEL | PRODUCT | ANIMAL | INGREDIENT)*>
<!ELEMENT CELL
  (PERSON | DEGREE | MODEL | PRODUCT | ANIMAL | INGREDIENT)*>
<!ELEMENT HEADING
  (PERSON | DEGREE | MODEL | PRODUCT | ANIMAL | INGREDIENT)*>
```

وتحتوي كل عناصر الحاوية على نفس المحتويات. وإذا ابتكرت عنصر جديد مثل EQUATION أو CD أو ACCOUNT، فيجب أن يتم إعلان هذه العناصر كتوابع ممكنة لعناصر الحاوية الثلاثة، وقد يسبب إضافتها إلى عنصرين، ولا مجال لإضافتها إلى ثلاثة، بعض المشكلات. ويزيد حجم المشكلة إذا كان لديك عناصر حاوية يبلغ عددها ٣٠ أو ٣٠٠ بدلاً من ٣.

ومن السهل جداً الاحتفاظ بـــ DTD إذا لم تقم بإعطاء كل حاوية قائمة توابع منفصلة خاصـــة بها. وبدلاً من ذلك يمكن جعل قائمة التوابع مرجع كينونة معامل، ثم اســـــتخدم مرجـــع كينونـــة المعامل هذا في كل من إعلانات عناصر الحاوية. فعلى سبيل المثال:

```
<!ENTITY % inlines
 ;(PERSON | DEGREE | MODEL | PRODUCT | ANIMAL |
INGREDIENT)*">
<!ELEMENT PARAGRAPH %inlines;>
<!ELEMENT CELL %inlines;>
<!ELEMENT HEADING %inlines;>
```

ولإضافة عنصر جديد، عليك فقط تغيير إعلان كينونة معامل واحد، بدلاً من تغييـر ٣ أو ٣٠ أو ٣٠٠ من إعلانات العناصر.

ولابد أن يتم إعلان مراجع كينونات المعاملات قبل استخدامها. وتعتبر الأمثلة التاليـــة غـــير صالحة حيث أن مرجع PCD% لم يعلن حتى استخدامه بالفعل مرتين:

```
<!ELEMENT FOOD %PCD;>
<!ELEMENT ANIMAL %PCD;>
<!ENTITY % PCD "(#PCDATA)">
```

ويمكن استخدام كينونات المعاملات لتوفير جزء من الإعلان في مجموعـــة DTD الفرعيـــة الخارجية. أي أن مرجع كينونات المعاملات قد تظهر فقط داخل إعـــلان فـــي مجموعـــة DTD الفرعية الخارجية. وتعتبر الأمثلة السابقة كلها غير صالحة إذا استخدمت فـــي مجموعـــة DTD فرعية داخلية.

ويمكن استخدام مجموعة DTD الفرعية الداخلية ومرجع كينونة المعامل خارج الإعلانـــات. فعلى سبيل المثال، يعتبر المثال التالي غير صالح في كل من مجموعتي DTD الفرعيتين الداخلية والخارجية:

```
<!ENTITY % hr "<!ELEMENT HR EMPTY>">
%hr;
```

وبالطبع لا يعتبر ذلك أسهل من إعلان عنصر HR بدون مراجع كينونات المعاملات:

```
<!ELEMENT HR EMPTY>
```

وسوف تستخدم مراجع كينونات المعاملات بصورة أساسية في مجموعـــات DTD الداخليـــة، عندما تشير هذه المراجع إلى كينونات معاملات خارجية، أي عند تجميعها فـــي مجموعـــات أو أجزاء من الإعلانات من ملف مختلف. وهو ما سنتناوله في القسم التالي.

كينونات المعاملات الخارجية

لقد استخدمنا في الأمثلة السابقة DTDs تقوم بتعريف كل العناصر المستخدمة في المستند. ولقد أصبحت هذه التقنية غير عملية مع المستندات الأكبر. وستحتاج أيضا إلى استخدام جزء من DTD في مواضع عديدة مختلفة.

فعلى سبيل المثال، إذا كان هناك DTD يصف عنوان بريدي، فإن تعريف العنوان يكون عاماً، ويمكن استخدامه بسهولة في سياقات عديدة مختلفة.

وبالمثل تفيد قائمة مراجع الكينونات التي سبق تعريفها والموجودة في تعليمات البرمجة ٩-٢ في معظم ملفات XML، ومن الأفضل ألا تقوم بنسخها ولصقها في كل مرة.

وتتيح لك كينونات المعاملات الخارجية إنشاء DTDs كبيرة من أخرى أصغر. أي أن أحد DTDs الخارجية قد يرتبط بأخر، وبفعل ذلك تتجذب العناصر والكينونات التي تم إعلانها في البداية. إلا أن DTD1 المحظور قد لا يشير إلى DTD2 إذا كان DTD2 يشير إلى DTD1، ويمكن أن تصبح DTDs أكبر وأكثر تعقيداً.

وفي نفس الوقت فإن تقسيم DTDs إلى أجزاء أصغر وأبسط يجعل تحليل DTD أكثر سهولة. ولقد كانت الأمثلة الموجودة في الفصل الأخير كبيرة بغير ضرورة وهذا بسبب تخزين مستند كامل وDTD الكامل الخاص به داخل ملف واحد. وعند تقسيم المستند وDTD الخاص به إلى ملفين منفصلين، يكون فهمهما أكثر سهولة.

وكذلك، فإن استخدام DTD أصغر وأكثر بساطة، يصف مجموعة واحدة من العناصر يجعل مزج ومطابقة DTDs التي قام أشخاص أو مؤسسات مختلفة بإنشائها، أكثر سهولة فعلى سبيل المثال، إذا كنت تكتب موضوعا عن توصيل درجات الحرارة المرتفعة، يمكنك استخدام DTD جزئية لوصف الأجزاء المضمنة. وعلم رياضيات DTD لكتابة معادلاتك، ورسوم اتجاهات DTD للأشكال وHTML DTD أساسي لمعالجة النص التوضيحي.



بصفة خاصة يمكن استخدام DTD mol.dtd من Peter Murray-Rust's Chemical Markup Language وDTD mathMl من W3C's Mathematical Markup Language وSVG DTD من W3C's Scalabel Vector Graphics وW3C's XHTML DTD.

وقد ترغب في أمثلة أكثر، حيث تحتاج إلى تركيب ومطابقات المفاهيم "ولذلك العلامات أيضاً" من حقول مختلفة. فالفكر الإنساني لا يتقيد بفئات معرفة ضيقة الحدود. فهو يتطلع إلى كل ما حوله. وسوف تعكس ذلك المستندات التي تقوم بكتابتها.

فلنرى كيفية تنظيم DTD لإحصاءات البيسبول كمركب من DTDs مختلفة ومن التقسـيمات الممكنة، كتابة DTDs منفصلة لكل من PLAYER وTEAM وSEASON. ويعتبر ذلك بعيداً عـن الطريقة الوحيدة لتقسيم DTD إلى أجزاء يمكن إدارتها بصورة أسهل، ولكنـها سـتكون مثـالاً مناسباً. وتوضح تعليمات البرمجة ٩-٥ DTD فقط للاعب يمكن تخزين ما يختص به بملف اسمه Player.dtd:

تعليمات البرمجة ٩-٥: DTD لعنصر PLAYER وتوابعه (Player.dtd)

```
<!- Player Info ->
 <!ELEMENT PLAYER (GIVEN_NAME, SURNAME, P, G,
   GS, AB?, R?, H?, D?, T?, HR?, RBI?, SB?, CS?,
   SH?, SF?, E?, BB?, S?, HBP?, W?, L?, SV?, CG?, SO?, ERA?,
   IP?, HRA?, RA?, ER?, HB?, WP?, B?, WB?, K?)
 >

 <!- Player's last name ->
 <!ELEMENT SURNAME (#PCDATA)>

  <!- Player's first name ->
 <!ELEMENT GIVEN_NAME (#PCDATA)>

  <!- Position ->
 <!ELEMENT P (#PCDATA)>

  <!-Games Played ->
 <!ELEMENT G (#PCDATA)>

<!-Games Started ->
 <!ELEMENT GS (#PCDATA)>


<!- ====================== ->
```

```
<!- Batting Statistics ->
<!- At Bats ->
<!ELEMENT AB (#PCDATA)>

<!- Runs ->
<!ELEMENT R (#PCDATA)>

<!- Hits ->
<!ELEMENT H (#PCDATA)>

<!- Doubles ->
<!ELEMENT D (#PCDATA)>

<!- Triples ->
<!ELEMENT T (#PCDATA)>

<!- Home Runs ->
<!ELEMENT HR (#PCDATA)>

<!- Runs Batted In ->
<!ELEMENT RBI (#PCDATA)>

<!- Stolen Bases ->
<!ELEMENT SB (#PCDATA)>

<!- Caught Stealing ->
<!ELEMENT CS (#PCDATA)>

<!- Sacrifice Hits ->
<!ELEMENT SH (#PCDATA)>

<!- Sacrifice Flies ->
```

```
<!ELEMENT SF (#PCDATA)>

<!- Errors ->
<!ELEMENT E (#PCDATA)>

<!- Walks (Base on Balls) ->
<!ELEMENT BB (#PCDATA)>

<!- Struck Out ->
<!ELEMENT S (#PCDATA)>

<!- Hit By Pitch ->
<!ELEMENT HBP (#PCDATA)>

<!- ========================->
<!- Pitching Statistics ->
<!- Complete Games ->
<!ELEMENT CG (#PCDATA)>

<!- Wins ->
<!ELEMENT W (#PCDATA)>

<!- Losses ->
<!ELEMENT L (#PCDATA)>

<!- Saves ->
<!ELEMENT SV (#PCDATA)>

<!- Shutouts ->
<!ELEMENT SO (#PCDATA)>

<!- ERA ->
```

```
<!ELEMENT ERA (#PCDATA)>

<!- Innings Pitched ->
<!ELEMENT IP (#PCDATA)>

<!- Home Runs hit Against ->
<!ELEMENT HRA (#PCDATA)>

<!- Runs hit Against ->
<!ELEMENT RA (#PCDATA)>

<!- Earned Runs ->
<!ELEMENT ER (#PCDATA)>

<!- Hit Batter ->
<!ELEMENT HB (#PCDATA)>

<!- Wild Pitches ->
<!ELEMENT WP (#PCDATA)>

<!- Balk ->
<!ELEMENT B (#PCDATA)>

<!- Walked Batter ->
<!ELEMENT WB (#PCDATA)>

<!- Struck Out Batter ->
<!ELEMENT K (#PCDATA)>

<!- ====================== ->
<!- Fielding Statistics ->
<!- Not yet supported ->
```

ولا يتيح لك DTD وحده إنشاء مستندات شديدة الأهمية. وتوضح تعليمات البرمجة ٩-٦ ملف صحيح بسيط يستخدم فقط PLAYER DTD الموجود في تعليمات البرمجة ٩-٥. ولا يعتبر هـذا الملف البسيط نفسه هاماً، إلا أنه يمكن إنشاء ملفات أخرى أكثر دقة خارج هذه الأجزاء الصغيرة.

تعليمات البرمجة ٩-٦: مستند صحيح باستخدام PLAYER DTD.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE PLAYER SYSTEM "player.dtd">
<PLAYER>
  <GIVEN_NAME>Chris</GIVEN_NAME>
  <SURNAME>Hoiles</SURNAME>
  <P>Catcher</P>
  <G>97</G>
  <GS>81</GS>
  <AB>267</AB>
  <R>36</R>
  <H>70</H>
  <D>12</D>
  <T>0</T>
  <HR>15</HR>
  <RBI>56</RBI>
  <SB>0</SB>
  <CS>1</CS>
  <SH>5</SH>
  <SF>4</SF>
  <E>3</E>
  <BB>38</BB>
  <S>50</S>
  <HBP>4</HBP>
</PLAYER>
```

ولكن ما هي الأجزاء الأخرى من المستند التي قد يكون لها DTDs الخاصة بها؟ بالطبع يعتبر TEAM جزء هام. ويمكن كتابة DTD الخاص به كما يلي:

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
<!ELEMENT TEAM_CITY (#PCDATA)>
<!ELEMENT TEAM_NAME (#PCDATA)>
```

إلا أنه عند قيامك باختبار أكثر قرباً من المفترض أن تلاحظ أن هناك شيئاً مفقوداً ألا وهـو: تعريف عنصر PLAYER. وهذا التعريف موجود في ملف Player.dtd المنفصل ويحتاج إلى أن يكون متصلا بهذا DTD.

ويتم اتصال DTDs بواسطة مراجع كينونات المعاملات الخارجية. وللحصـول علـى DTD خاص، يأخذ الاتصال هذا الشكل التالي:

```
<!ENTITY % name SYSTEM "URI">
%name;
```

فعلي سبيل المثال:

```
<!ENTITY % player SYSTEM "player.dtd">
%player;
```

ويستخدم هذا المثال URL نسبي (Player .dtd) ويفترض أن ملـف Player .dtd سـوف يتواجد في نفس موضع DTD المرتبط. وإذا لم يكن هذا هو ما يحدث، يمكنـك اسـتخدام URL كامل كما يلي:

```
<!ENTITY % player SYSTEM
  "http://metalab.unc.edu/xml/dtds/player.dtd">
%player;
```

وتوضح تعليمات البرمجة ٩-٧ TEAM DTD كامل يشتمل على مرجع PLAYER DTD:

تعليمات البرمجة ٩-٧: TEAM DTD (team .dtd)

```
<!ELEMENT TEAM (TEAM_CITY, TEAM_NAME, PLAYER*)>
<!ELEMENT TEAM_CITY (#PCDATA)>
<!ELEMENT TEAM_NAME (#PCDATA)>
<!ENTITY % player SYSTEM "player.dtd">
%player;
```

ويحتـوي SEASON علـى عنـاصر LEAGUE وDIVISION وTEAM إلا أن LEAGUE وDIVISION قد يحتوي كل منهما على DTD الخاص به. ولا يفيد تقسـيم DTDs إلـى أبعـد الحدود. فإذا لم يكن لديك بعض المستندات التي تحتوي على عنصـر LEAGUE أو DIVISION

اللذان لا يعتبران جزءاً من SEASON، فقد تقوم أيضاً بتضمين العناصر الثلاثة كلها بنفس DTD. وهذا ما توضحه تعليمات البرمجة ٩-٨.

تعليمات البرمجة ٩-٨: SEASON DTD (season.dtd)

```
<!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT LEAGUE (LEAGUE_NAME, DIVISION, DIVISION,
DIVISION)>

  <!- American or National ->
  <!ELEMENT LEAGUE_NAME (#PCDATA)>

  <!- East, West, or Central ->
  <!ELEMENT DIVISION_NAME (#PCDATA)>
  <!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
  <!ELEMENT SEASON (YEAR, LEAGUE, LEAGUE)>
  <!ENTITY % team SYSTEM "team.dtd">
  %team;
```

إنشاء مستند يتكون من أجزاء

لقد كانت الأمثلة الخاصة بالبيسبول كثيرة جداً. إلا أن ما يظهر في هذا الكتاب هو فقط إصدار مقطوع من أعداد لاعبين محددة، ويبلغ المستند الكامل أكثر من نصف ميجابايت، وهو يعتبر كبير جدا بالنسبة للتحميل أو للبحث به، خاصة إذا كان القارئ مهتما بفريق أو لاعب أو تقسيم واحد فقط. وتتيح لك التقنيات الموضحة في القسم السابق من هذا الفصل تقسيم المستند إلى مستندات عديدة مختلفة أصغر حجما وأكثر سهولة، على أساس مستند واحد لكل فريق وكل لاعب وكل تقسيم وكل مجموعة. وتقوم مراجع الكينونات الخارجية بتجميع اللاعبين لتشكيل فرق وتجميع الفرق لتكوين تقسيمات. والتقسيمات لتشكيل مجموعات. والمجموعات لتشكيل فترات.

ومن المؤسف أنك لا تستطيع تضمين أي مستند XML ككينونة خارجية موزعة. فانظر، على سبيل المثال، إلى تعليمات البرمجة ٩-٩، ChrisHoiles.xml. وهي تعتبر إصدار معدل من تعليمات البرمجة ٩-٦. إلا أنك إذا دققت أكثر، ستلاحظ اختلاف التسجيل الأولي. والتسجيل الأولي لتعليمات البرمجة ٩-٦ هو:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE PLAYER SYSTEM "player.dtd">
```

والتسجيل الأولي لتعليمات البرمجة ٩-٩ هو ببساطة إعلان XML بدون سمة Standalone بينما يحتوي على سمة encoding. وكذلك فقد تم تجاهل إعلان نوع المستند تماماً. وفي ملف مثل الموضح في تعليمات البرمجة ٩-٩ والمراد تضمينه داخل مستند أخر، يطلق على هذا الفرز لإعلان XML: *text declaration* "أو إعلان النص"، بالرغم من أنه، كما ترى، بالفعل إعلان XML صحيح.

تعليمات البرمجة ٩-٩: ChrisHoiles.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<PLAYER>
  <GIVEN_NAME>Chris</GIVEN_NAME>
  <SURNAME>Hoiles</SURNAME>
  <P>Catcher</P>
  <G>97</G>
  <GS>81</GS>
  <AB>267</AB>
  <R>36</R>
  <H>70</H>
  <D>12</D>
  <T>0</T>
  <HR>15</HR>
  <RBI>56</RBI>
  <SB>0</SB>
  <CS>1</CS>
  <SH>5</SH>
  <SF>4</SF>
  <E>3</E>
  <BB>38</BB>
  <S>50</S>
  <HBP>4</HBP>
</PLAYER>
```

ولابد أن تكون لإعلانات النص سمة encoding "على عكس الإعلانات التي قد، وليس من الضروري أن، يكون لها سمة encoding" التي تحدد مجموعة الأحرف التي تستخدمها الكينونة. ويتيح ذلك للمستندات المركبة أن تتكون من كينونات في مجموعات أحرف مختلفة. فعلى سبيل المثال، قد يتم جمع مستند في Latin-5 مع مستند في UTF-8. وينبغي أن يفهم المعالج/المستعرض كل ما هو مكتوب على هيئة تعليمات برمجية المستخدمة عن طريق كينونات مختلفة.

وكل الأمثلة المذكورة في هذا الفصل موجودة في ASCII. وبينما ASCII هي مجموعة قيود فرعية لكل من ISO Latin-1 وUTF-8، فبإمكانك رؤية أي من إعلاني النص التاليين:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml version="1.0" encoding="UTF-8"?>
```

وتوضح كل من تعليمات البرمجة ٩-١٠، mets.dtd، وتعليمات البرمجة ٩-١١، mets.xml. كيف يمكن استخدام الكينونات الخارجية الموزعة لتكون معاً فريقاً متكامل. ويقوم مستند XML بتحميل DTD باستخدام مرجع كينونة معامل خارجي في مجموعة DTD الفرعية الداخلية الخاصة به. حيث تشتمل كينونة المستند على مراجع كينونات خارجية عامة متعددة التي تقوم بتحميل اللاعبين الفرديين.

تعليمات البرمجة ٩-١٠: New York Mets DTD تحتوي على مراجع الكينونات الخاصة باللاعبين (mets.dtd)

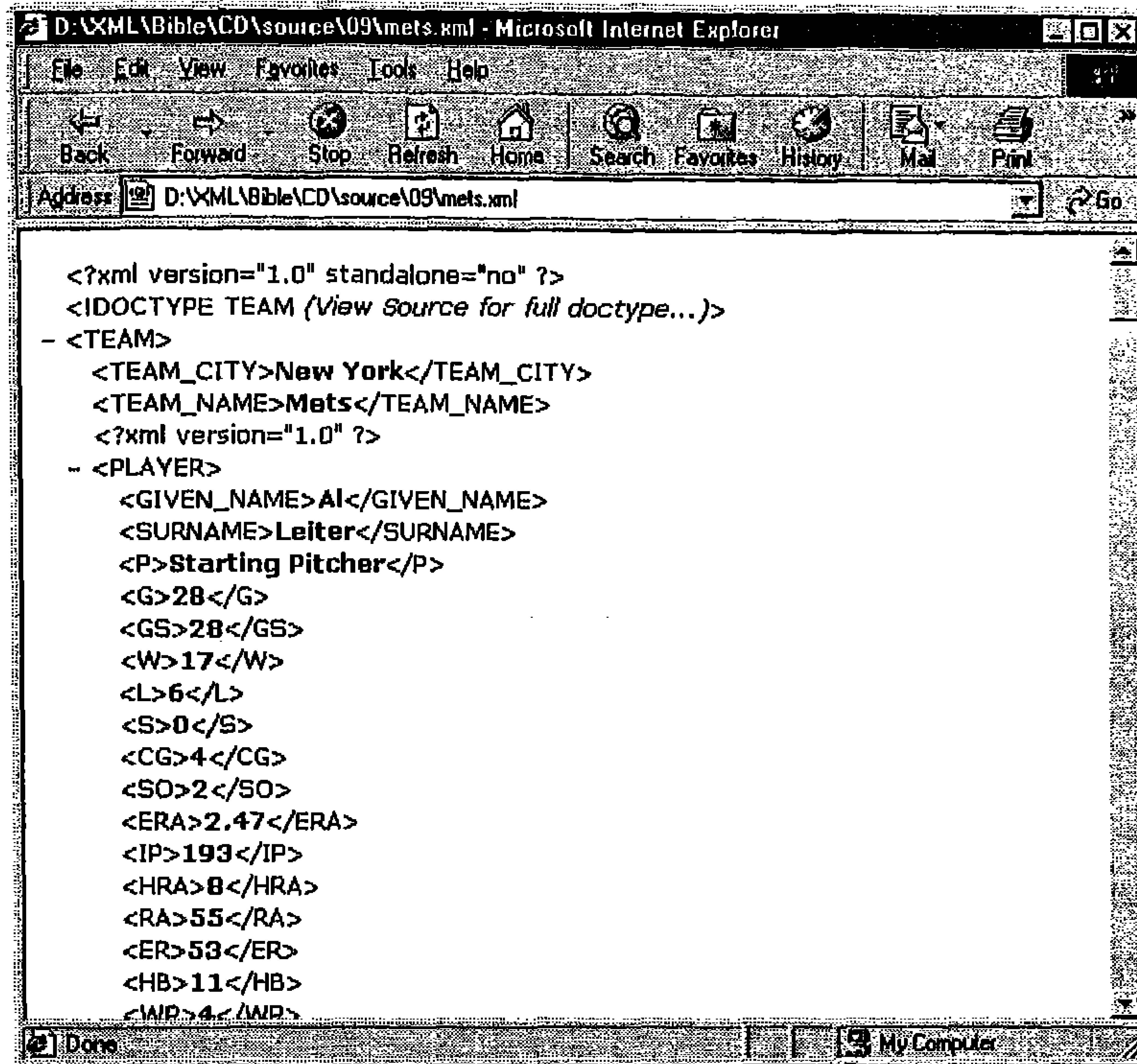
```
<!ENTITY AlLeiter SYSTEM "mets/AlLeiter.xml">
<!ENTITY ArmandoReynoso SYSTEM "mets/ArmandoReynoso.xml">
<!ENTITY BobbyJones SYSTEM "mets/BobbyJones.xml">
<!ENTITY BradClontz SYSTEM "mets/BradClontz.xml">
<!ENTITY DennisCook SYSTEM "mets/DennisCook.xml">
<!ENTITY GregMcmichael SYSTEM "mets/GregMcmichael.xml">
<!ENTITY HideoNomo SYSTEM "mets/HideoNomo.xml">
<!ENTITY JohnFranco SYSTEM "mets/JohnFranco.xml">
<!ENTITY JosiasManzanillo SYSTEM "mets/JosiasManzanillo.xml">
<!ENTITY OctavioDotel SYSTEM "mets/OctavioDotel.xml">
<!ENTITY RickReed SYSTEM "mets/RickReed.xml">
```

```
<!ENTITY RigoBeltran SYSTEM "mets/RigoBeltran.xml">
<!ENTITY WillieBlair SYSTEM "mets/WillieBlair.xml">
```

ويعرض الشكل ٩-٣ مستند XML عند تحميله داخل Internet Explorer. ولاحظ أن كل البيانات الخاصة بجميع اللاعبين قد تم عرضها حتى بالرغم من أن المستند الأساسي يحتوي فقط على مراجع إلى الكينونات حيث تتواجد بيانات اللاعبين. ويقوم Internet Explorer بحل المراجع الخارجية، ولا تقوم كل موزعات/ مستعرضات XML بذلك.

تعليمات البرمجة ٩-١١: New York Mets تحتوي على لاعبين ثم تحميل ما يختص بهم من كينونات خارجية (mets.xml)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE TEAM SYSTEM "team.dtd" [
  <!ENTITY % players SYSTEM "mets.dtd">
  %players;
]>
<TEAM>
  <TEAM_CITY>New York</TEAM_CITY>
  <TEAM_NAME>Mets</TEAM_NAME>
  &AlLeiter;
  &ArmandoReynoso;
  &BobbyJones;
  &BradClontz;
  &DennisCook;
  &GregMcmichael;
  &HideoNomo;
  &JohnFranco;
  &JosiasManzanillo;
  &OctavioDotel;
  &RickReed;
  &RigoBeltran;
  &WillieBlair;
</TEAM>
```



الشكل ٩-٣ مستند XML عند عرض كل اللاعبين في New York mets ١٩٩٨.

ومن الأفضل الاستمرار في هذا الإجراء وإنشاء تقسيم بواسطة دمج ملفات الفرق، ومجموعات بواسطة دمج التقسيمات وفترات بواسطة دمج المجموعات. ولسوء الحظ، إذا حاولت ذلك فستجد الأمر صعباً. فالمستندات التي يتم تضمينها عن طريق كينونات خارجية لا يمكن أن تكون لها DTDs الخاصة بها. وغالبا ما قد يحتوي التسجيل الأولي الخاص بها على إعلان نصي. وهذا يعني أن بإمكانك أن يكون لديك مستوى واحد من تضمين المستند. ويتقيد ذلك بتضمين DTD حيث يمكن تضمين DTDs عشوائياً.

لذلك فإن خيارك المحتمل الوحيد هو تضمين كل الفرق والتقسيمات والمجموعات والفترات بمستند واحد يشير إلى مستندات اللاعبين العديدة. ويتطلب ذلك ما هو أكثر قليلاً من إعلانات كينونات ١٢٠٠ "أي واحد لكل لاعب". وبينما يمكن أن تقوم DTDs بالتضمين عشوائياً، ستكون بدايتنا هي DTD الذي يجمع DTDs كما في تعليمات البرمجة ٩-١٠ التي تحتوي على تعريفات كينونات لكل الفرق. وهو ما توضحه تعليمات البرمجة ٩-١٢.

تعليمات البرمجة ٩-١٢: DTD للاعبين (Players.dtd)

```
<!ENTITY % angels SYSTEM "angels.dtd">
%angels;
<!ENTITY % astros SYSTEM "astros.dtd">
%astros;
<!ENTITY % athletics SYSTEM "athletics.dtd">
%athletics;
<!ENTITY % bluejays SYSTEM "bluejays.dtd">
%bluejays;
<!ENTITY % braves SYSTEM "braves.dtd">
%braves;
<!ENTITY % brewers SYSTEM "brewers.dtd">
%brewers;
<!ENTITY % cubs SYSTEM "cubs.dtd">
%cubs;
<!ENTITY % devilrays SYSTEM "devilrays.dtd">
%devilrays;
<!ENTITY % diamondbacks SYSTEM "diamondbacks.dtd">
%diamondbacks;
<!ENTITY % dodgers SYSTEM "dodgers.dtd">
%dodgers;
<!ENTITY % expos SYSTEM "expos.dtd">
%expos;
<!ENTITY % giants SYSTEM "giants.dtd">
%giants;
<!ENTITY % indians SYSTEM "indians.dtd">
%indians;
<!ENTITY % mariners SYSTEM "mariners.dtd">
%mariners;
<!ENTITY % marlins SYSTEM "marlins.dtd">
```

```
%marlins;
<!ENTITY % mets SYSTEM "mets.dtd">
%mets;
<!ENTITY % orioles SYSTEM "orioles.dtd">
%orioles;
<!ENTITY % padres SYSTEM "padres.dtd">
%padres;
<!ENTITY % phillies SYSTEM "phillies.dtd">
%phillies;
<!ENTITY % pirates SYSTEM "pirates.dtd">
%pirates;
<!ENTITY % rangers SYSTEM "rangers.dtd">
%rangers;
<!ENTITY % redsox SYSTEM "redsox.dtd">
%redsox;
<!ENTITY % reds SYSTEM "reds.dtd">
%reds;
<!ENTITY % rockies SYSTEM "rockies.dtd">
%rockies;
<!ENTITY % royals SYSTEM "royals.dtd">
%royals;
<!ENTITY % tigers SYSTEM "tigers.dtd">
%tigers;
<!ENTITY % twins SYSTEM "twins.dtd">
%twins;
<!ENTITY % whitesox SYSTEM "whitesox.dtd">
%whitesox;
<!ENTITY % yankees SYSTEM "yankees.dtd">
%yankees;
```

وتقوم تعليمات البرمجة ٩-١٣، أي المستند الرئيسي، بتجميع كل مستندات اللاعبين الفرعية وDTDs التي تقوم بتعريف كينونات كل لاعب. وبالرغم من أن هذا المستند أصغر كثيرا من

المستند الذي تم تطويره فيما سبق "٣٢ كيلو في مقابل ٦٢٨ كيلو" فإنه لا يزال طويل جداً، لذلك فلم يتم تضمين كل اللاعبين هنا. والإصدار الكامل لتعليمات البرمجة ٩-١٣ يقوم على ٣٣ DTDs وما يزيد عن ١٠٠٠ ملف XML لإنتاج المستند المتكامل. والمشكلة الكبيرة المصاحبة لهذه الطريقة هي أنها تتطلب ما يزيد عن ١٠٠٠ اتصال منفصل بخادم الويب قبل أن يتاح عرض المستند.

تعليمات البرمجة ٩-١٣: المستند الرئيسي الخاص بفترة ١٩٩٨ يستخدم مراجع كينونة خارجية للاعبين.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE SEASON SYSTEM "baseball.dtd" [
<!ENTITY % players SYSTEM "players.dtd">
  %players;

]>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>National</LEAGUE_NAME>
    <DIVISION>
      <DIVISION_NAME>East</DIVISION_NAME>
       <TEAM>
         <TEAM_CITY>Florida</TEAM_CITY>
         <TEAM_NAME>Marlins</TEAM_NAME>
       </TEAM>
       <TEAM>
         <TEAM_CITY>Montreal</TEAM_CITY>
         <TEAM_NAME>Expos</TEAM_NAME>
</TEAM>
       <TEAM>
         <TEAM_CITY>New York</TEAM_CITY>
         <TEAM_NAME>Mets</TEAM_NAME>
```

```
        &RigoBeltran;
        &DennisCook;
        &SteveDecker;
        &JohnFranco;
        &MattFranco;
        &ButchHuskey;
        &BobbyJones;
        &MikeKinkade;
        &HideoNomo;
        &VanceWilson;
    </TEAM>
    <TEAM>
      <TEAM_CITY>Philadelphia</TEAM_CITY>
      <TEAM_NAME>Phillies</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Chicago</TEAM_CITY>
      <TEAM_NAME>Cubs</TEAM_NAME>
    </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    <TEAM>
      <TEAM_CITY>Arizona</TEAM_CITY>
      <TEAM_NAME>Diamondbacks</TEAM_NAME>
    </TEAM>
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
```

```
    <DIVISION_NAME>East</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Baltimore</TEAM_CITY>
       <TEAM_NAME>Orioles</TEAM_NAME>
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Chicago</TEAM_CITY>
       <TEAM_NAME>White Sox</TEAM_NAME>
         &JeffAbbott;
         &MikeCameron;
         &MikeCaruso;
         &LarryCasian;
         &TomFordham;
         &MarkJohnson;
         &RobertMachado;
         &JimParque;
         &ToddRizzo;
     </TEAM>
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
     <TEAM>
       <TEAM_CITY>Anaheim</TEAM_CITY>
       <TEAM_NAME>Angels</TEAM_NAME>
     </TEAM>
  </DIVISION>
 </LEAGUE>
</SEASON>
```

ويتاح لك بعض المرونة تختار بها مستويات لمستندك الرئيسي والبيانات المضمنة، فعلى سبيل المثال من البدائل المتاحة للبنية المستخدمة بواسطة تعليمات البرمجة ٩-١٢ هي وضع

الفرق وكل لاعبيها داخل مستندات فردية ثم ضم ملفات الفرق هذه بفترة باستخدام كينونات خارجية كما هو موضح بتعليمات برمجة ٩-١٤. ومن إحدى مميزات ذلك هي استخدام عدد أقل من ملفات XML لأحجام أكثر تضع تحميلاً أقل على خادم الويب وسنقوم بالتحميل والعرض بصورة أكثر سرعة. وللأمانة، لا تتميز إحدى الطريقتين على الأخرى. فلك حرية استخدام الطريقة التي تشعر أنها تتطابق التنظيم الخاص ببياناتك، أو الطريقة التي تشعر بالراحة أكثر في استخدامها.

تعليمات البرمجة ٩-١٤ : فترة ١٩٩٨ تستخدم مراجع كينونات عامة للفرق

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE SEASON SYSTEM "baseball.dtd" [
<!ENTITY angels SYSTEM "angels.xml">
  <!ENTITY astros SYSTEM "astros.xml">
  <!ENTITY athletics SYSTEM "athletics.xml">
  <!ENTITY bluejays SYSTEM "bluejays.xml">
  <!ENTITY braves SYSTEM "braves.xml">
  <!ENTITY brewers SYSTEM "brewers.xml">
  <!ENTITY cubs SYSTEM "cubs.xml">
<!ENTITY devilrays SYSTEM "devilrays.xml">
  <!ENTITY diamondbacks SYSTEM "diamondbacks.xml">
  <!ENTITY dodgers SYSTEM "dodgers.xml">
  <!ENTITY expos SYSTEM "expos.xml">
  <!ENTITY giants SYSTEM "giants.xml">
  <!ENTITY indians SYSTEM "indians.xml">
  <!ENTITY mariners SYSTEM "mariners.xml">
  <!ENTITY marlins SYSTEM "marlins.xml">
  <!ENTITY mets SYSTEM "mets.xml">
  <!ENTITY orioles SYSTEM "orioles.xml">
  <!ENTITY padres SYSTEM "padres.xml">
  <!ENTITY phillies SYSTEM "phillies.xml">
  <!ENTITY pirates SYSTEM "pirates.xml">
  <!ENTITY rangers SYSTEM "rangers.xml">
  <!ENTITY redsox SYSTEM "red sox.xml">
```

```
<!ENTITY reds SYSTEM "reds.xml">
<!ENTITY rockies SYSTEM "rockies.xml">
<!ENTITY royals SYSTEM "royals.xml">
<!ENTITY tigers SYSTEM "tigers.xml">
<!ENTITY twins SYSTEM "twins.xml">
<!ENTITY whitesox SYSTEM "whitesox.xml">
<!ENTITY yankees SYSTEM "yankees.xml">

]>
<SEASON>
  <YEAR>1998</YEAR>
  <LEAGUE>
    <LEAGUE_NAME>National</LEAGUE_NAME>
    <DIVISION>
      <DIVISION_NAME>East</DIVISION_NAME>
       &marlins;
       &braves;
       &expos;
       &mets;
       &phillies;
    </DIVISION>
    <DIVISION>
      <DIVISION_NAME>Central</DIVISION_NAME>
       &cubs;
       &reds;
       &astros;
       &brewers;
       &pirates;
    </DIVISION>
    <DIVISION>
      <DIVISION_NAME>West</DIVISION_NAME>
      &diamondbacks;
      &rockies;
      &dodgers;
```

```
    &padres;
    &giants;
  </DIVISION>
</LEAGUE>
<LEAGUE>
  <LEAGUE_NAME>American</LEAGUE_NAME>
  <DIVISION>
    <DIVISION_NAME>East</DIVISION_NAME>
    &orioles;
    &redsox;
    &yankees;
    &devilrays;
    &bluejays
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>Central</DIVISION_NAME>
    &whitesox;
    &indians;
    &tigers;
    &royals;
    &twins;
  </DIVISION>
  <DIVISION>
    <DIVISION_NAME>West</DIVISION_NAME>
    &angels;
    &athletics;
    &mariners;
    &rangers;
  </DIVISION>
</LEAGUE>
</SEASON>
```

وهناك بديل أخر أقل احتمالاً وهو إنشاء فرق فعليا من كينونات لاعبين خارجية داخل ملفات منفصلة ثم ضم ملفات الفرق هذه داخل تقسيمات ومجموعات وفترات. ويمكن للمستند الرئيسي

تعريف مراجع كينونات مستخدمة في مستندات الفرق التابعة. إلا أنه في هذه الحالة لا تكون ملفات الفرق قابلة للاستخدام حيث أن مراجع الكينونات لا يتم تعريفها حتى يتم تجميعها داخل المستند الرئيسي.

ومن المؤسف حقا أن المستند ذو المستوى الأعلى هو فقط الذي يمكن إرفاقه في DTD. وهو يحد من فائدة الكينونات الخارجية الموزعة. إلا أنك عندما تتعرف أكثر على X Links وX Pointers، سوف تتعرف على بعض الطرق الأخرى لإنشاء مستندات كبيرة مركبة من أجزاء صغيرة. إلا هذه التقنيات لا تعتبر جزء من معيار XML الأساسي وغير مدعمة بصورة جيدة عن طريق أي من معالجات XML للتحقق من الصحة أو مستعرضات الويب مثل التقنيات المذكورة في هذا الفصل.



يغطي الفصل ١٦ "X Links" بينما يتناول الفصل ١٧ "Xpionters".

الكينونات وتعريفات نوع المستند في المستندات صحيحة التكوين

يتناول الجزء الأول من هذا الكتاب مستندات XML صحيحة التكوين بدون DTDs. بينما يتناول الجزء الثاني المستندات التي لها DTDs وتلتزم بالقيود الموجودة في DTD، ألا وهي المستندات الصحيحة. إلا أنه يوجد مستوى ثالث للمطابقة لمعيار XML ألا وهو: المستندات التي لها DTDs وتكون صحيحة التكوين وليست صالحة، إما بسبب عدم اكتمال DTD أو أن المستند لا يلتزم بقيود DTD. ويعتبر هذا النوع أقل الأنواع الثلاثة شيوعاً.

إلا أنه ليس من الضروري أن تكون كل المستندات صحيحة. ففي بعض الأحيان يكفي فقط أن يكون مستند XML صحيح التكوين. وهناك موضع يخص DTDs في مستندات XML صحيحة التكوين "بالرغم من عدم ضرورة كونها مستندات صالحة". وهناك بعض معالجات XML التي تقوم بالتحقق من الصحة تستفيد من المعلومات الموجودة في DTD دون أن تتطلب مطابقة تامة لها.

وإذا احتوى مستند صحيح التكوين ولكن غير صالح على DTD، فينبغي أن يكون DTD هذا له نفس الشكل العام، كما سبق توضيح ذلك في الفصول السابقة. أي أنه يبدأ بإعلان نوع المستند وقد يحتوي على إعلانات ELEMENT وATTLIST وENTITY. ويختلف مثل هذا المستند عن المستند الصالح، الذي يهتم فيه المعالج فقط بإعلانات ENTITY.

الكينونات الداخلية

وتعتبر أهم مميزات استخدام DTD داخل مستند XML صحيح التكوين وغير صالح هي أنك قد تستخدم مراجع الكينونات العامة الداخلية بجانب المراجع المعرفة سابقاً وهي < و> و" و' و&. ويمكن الإعلان عن الكينونات التي تريد استخدامها كالمعتاد ثم استخدامها في مستندك.

فعلى سبيل المثال، لتكرار المثال السابق، بفرض أنك تريد استبدال مرجع الكينونة &ERH; بسلسلة "Elliotte Rusty Harold"، إلا أنك لا ترغب في كتابة DTD كاملاً لمستندك. قم ببساطة بالإعلان عن مرجع الكينونة ERH في DTD، كما توضح تعليمات البرمجة ٩-١٥. ويعتبر هذا المستند صحيح التكوين فقط وليس صالحاً، إلا أنه يمكن قبوله إذا لم تكن تتطلب التحقق من الصحة.

تعليمات البرمجة ٩-١٥: مرجع كينونة ERH في DTD يؤدي إلى مستند صحيح التكوين وليس صالحاً بعد

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY ERH "Elliotte Rusty Harold">
]>
<DOCUMENT>
  <TITLE>&ERH;</TITLE>
  <SIGNATURE>
    <COPYRIGHT>1999 &ERH;</COPYRIGHT>
    <EMAIL>elharo@metalab.unc.edu</EMAIL>
    <LAST_MODIFIED>March 10, 1999</LAST_MODIFIED>
  </SIGNATURE>
</DOCUMENT>
```

ويعتبر إعلان نوع المستند الموجود في تعليمات البرمجة ٩-١٥ متناثراً بصورة شديدة. وبجانب تعريف مرجع الكينونة ERH، فهو يوضح ببساطة أن العنصر الجذري هو Document إلا عدم صحة التكوين لا تتطلب التزام المستند بهذا القيد الصغير. فعلى سبيل المثال، تعرض تعليمات البرمجة ٩-١٦ مستند أخر يستخدم عنصر PAGE الجذري حتى بالرغم من أن إعلان

نوع المستند يوضح أنه من المفترض أن يكون العنصر الجذري. ولا يزال هذا المستند صحيح التكوين ولكن ليس صالحاً، مثلما كان الأمر مع تعليمات البرمجة ٩-١٥.

تعليمات البرمجة ٩-١٦: مستند صحيح التكوين وفي نفس الوقت غير صالح

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY ERH "Elliotte Rusty Harold">
]>
<PAGE>
  <TITLE>&ERH;</TITLE>
  <SIGNATURE>
    <COPYRIGHT>1999 &ERH;</COPYRIGHT>
    <EMAIL>elharo@metalab.unc.edu</EMAIL>
    <LAST_MODIFIED>March 10, 1999</LAST_MODIFIED>
   </SIGNATURE>
</PAGE>
```

ويمكن أن يحتوي DTD على إعلانات <!ELEMENT> و<!ATTLIST> و<!NOTATION> الأخرى أيضاً. وتتجاهل المعالجات التي لا تقوم بالتحقق من الصحة هذه الإعلانات كلها. ولكنها تهتم فقط بإعلانات <!ENTITY> ويتعارض DTD الموجود في تعليمات البرمجة ٩-١٧ مع محتوياته بصورة واضحة. فعلى سبيل المثال، من المفترض أن يكون عنصر ADDRESS فارغاً بحسب DTD، ولكن في الواقع يحتوي على عناصر تابعة عديدة، غير معلن عنها. ومن المفترض أيضاً أن يكون لكل عنصر ADDRESS سمات OCCUPANT وSTREET وCITY وZIP، إلا أنها لا تكون ظاهرة ومن المفترض أن يكون العنصر الجذري هو DOCUMENT وليس ADRESS. وينبغي أن يحتوي عنصر DOCUMENT على TITLE وSIGNATUR، ولا يتم إعلان أي منهما في DTD. ولا يزال هذا المستند صحيح التكوين وكذلك صالح جداً.

تعليمات البرمجة ٩-١٧: مستند صالح وفي نفس الوقت صحيح التكوين

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY ERH "Elliotte Rusty Harold">
  <!ELEMENT ADDRESS EMPTY>
  <!ELEMENT DOCUMENT (TITLE, ADDRESS+, SIGNATURE)>
  <!ATTLIST ADDRESS OCCUPANT   CDATA #REQUIRED>
  <!ATTLIST ADDRESS DEPARTMENT CDATA #IMPLIED>
  <!ATTLIST ADDRESS COMPANY    CDATA #IMPLIED>
  <!ATTLIST ADDRESS STREET     CDATA #REQUIRED>
  <!ATTLIST ADDRESS CITY       CDATA #REQUIRED>
  <!ATTLIST ADDRESS ZIP        CDATA #REQUIRED>
]>
<ADDRESS>
  <OCCUPANT>Elliotte Rusty Harold</OCCUPANT>
  <DEPARTMENT>Computer Science</DEPARTMENT>
  <COMPANY>Polytechnic University</COMPANY>
  <STREET>5 Metrotech Center</STREET>
  <CITY>Brooklyn</CITY>
  <STATE>NY</STATE>
  <ZIP>11201</ZIP>
</ADDRESS>
```

الكينونات الخارجية

قد تقوم المعالجات التي لا تتحقق من صحة المستندات بحل مراجع الكينونات الخارجية، إلا أنها ليست مطالبة بذلك. فعلى سبيل المثال، لا يقوم موزع مصدر XML المفتوح والمستخدم بواسطة Mozilla بحل مراجع الكينونات الخارجية. إلا أن معظم المعالجات الأخرى التي تشتمل على الموزع المستخدم في Internet Explorer تقوم بذلك. وقد تقوم المعالجات التي لا تتحقق من الصحة بحل الكينونات غير الموزعة التي تحتوي على بيانات غير XML مثل الصور والأصوات.

وتفيد الكينونات الخارجية على الأخص في تخزين النص المتداول. فعلـــى ســـبيل المثـــال، HTML تقوم بتعريف مراجع الكينونات للحروف غير ASCII ISO Latin-1 التي تعتبر أســـهل بعض الشيء لتذكرها من مراجع كينونات الأحرف العددية. فعلى سبيل المثال، تكـــون å هـــي ˚ وþ هي þ وÝ هي Ý وهكذا. وتعتبر تعليمات البرمجـــة ٩-١٨ هـــي ISO DTD الرسمي الذي يقوم بتعريف هذه المراجع "مع تعديل بســـيط للتعليقـــات والمســـافات البيضاء لجعله يتلاءم بصورة محكمة داخل الصفحة".

تعليمات البرمجة ٩-١٨: DTD الخاص بالأحرف غير ASCII ISO-LATIN-1

```
<!- (C) International Organization for Standardization 1986
   Permission to copy in any form is granted for use with
   conforming SGML systems and applications as defined in
   ISO 8879, provided this notice is included in all copies.
->

<!- Character entity set. Typical invocation:
   <!ENTITY % ISOlat1 PUBLIC
     "ISO 8879-1986//ENTITIES Added Latin 1//EN//XML">
   %ISOlat1;
->

<!- This version of the entity set can be used with any SGML
   document which uses ISO 8859-1 or ISO 10646 as its
   document character set. This includes XML documents and
   ISO HTML documents.

  Version: 1998-10-01
->


<!ENTITY Agrave "&#192;"><!- capital A, grave accent ->
<!ENTITY Aacute "&#193;"><- capital A, acute accent ->
<!ENTITY Acirc  "&#194;"><!- capital A, circumflex accent ->
```

```
<!ENTITY Atilde "&#195;"><!- capital A, tilde ->
<!ENTITY Auml   "&#196;"><!- capital A, dieresis umlaut ->
<!ENTITY Aring  "&#197;"><!- capital A, ring ->
<!ENTITY AElig  "&#198;"><!- capital AE diphthong ligature->
<!ENTITY Ccedil "&#199;"><!- capital C, cedilla ->
<!ENTITY Egrave "&#200;"><!- capital E, grave accent ->
<!ENTITY Eacute "&#201;"><!- capital E, acute accent ->
<!ENTITY Ecirc  "&#202;"><!- capital E, circumflex accent ->
<!ENTITY Euml   "&#203;"><!- capital E, dieresis umlaut ->
<!ENTITY Igrave "&#204;"><!- capital I, grave accent ->
<!ENTITY Iacute "&#205;"><!- capital I, acute accent ->
<!ENTITY Icirc "&#206;" ><!- capital I, circumflex accent ->
<!ENTITY Iuml   "&#207;" ><!- capital I, dieresis umlaut ->
<!ENTITY ETH    "&#208;" ><!- capital Eth, Icelandic ->
<!ENTITY Ntilde "&#209;" ><!- capital N, tilde ->
<!ENTITY Ograve "&#210;" ><!- capital O, grave accent ->
<!ENTITY Oacute "&#211;" ><!- capital O, acute accent ->
<!ENTITY Ocirc  "&#212;" ><!- capital O, circumflex accent ->
<!ENTITY Otilde "&#213;" ><!- capital O, tilde ->
<!ENTITY Ouml   "&#214;" ><!-capital O dieresis/umlaut mark->
<!ENTITY Oslash "&#216;" ><!- capital O, slash ->
<!ENTITY Ugrave"&#217;" ><!- capital U, grave accent ->
<!ENTITY Uacute "&#218;" ><!- capital U, acute accent ->
<!ENTITY Ucirc "&#219;" ><!- capital U circumflex accent ->
<!ENTITY Uuml   "&#220;" ><!- capital U dieresis umlaut ->
<!ENTITY Yacute "&#221;" ><!- capital Y, acute accent ->
```

```
<!ENTITY THORN  "&#222;" ><!- capital THORN, Icelandic ->
<!ENTITY szlig  "&#223;" ><!- small sharp s, (sz ligature) ->
<!ENTITY agrave "&#224;" ><!- small a, grave accent ->
<!ENTITY aacute "&#225;" ><!- small a, acute accent ->
<!ENTITY acirc  "&#226;"><!- small a, circumflex accent ->
<!ENTITY atilde "&#227;"><!- small a, tilde ->
<!ENTITY auml   "&#228;"><!- small a dieresis/umlaut mark->
<!ENTITY aring  "&#229;" ><!- small a, ring ->
<!ENTITY aelig  "&#230;" ><!- small ae, diphthong ligature ->
<!ENTITY ccedil "&#231;" ><!- small c, cedilla ->
<!ENTITY egrave "&#232;" ><!- small e, grave accent ->
<!ENTITY eacute "&#233;" ><!- small e, acute accent ->
<!ENTITY ecirc  "&#234;"><!- small e, circumflex accent ->
<!ENTITY euml   "&#235;"><!- small e, dieresis or umlaut ->
<!ENTITY igrave "&#236;" ><!- small i, grave accent ->
<!ENTITY iacute "&#237;" ><!- small i, acute accent ->
<!ENTITY icirc  "&#238;"><!- small i, circumflex accent ->
<!ENTITY iuml   "&#239;"><!- small i, dieresis or umlaut ->
<!ENTITY eth    "&#240;" ><!- small eth, Icelandic ->
<!ENTITY ntilde  "&#241;"><!- small n, tilde ->
<!ENTITY ograve  "&#242;"><!- small o, grave accent ->
<!ENTITY oacute  "&#243;"><!- small o, acute accent ->
<!ENTITY ocirc   "&#244;" ><!- small o, circumflex accent ->
<!ENTITY otilde  "&#245;" ><!- small o, tilde ->
<!ENTITY ouml    "&#246;"><!- small o, dieresis or umlaut->
<!ENTITY oslash "&#248;"><!- small o, slash ->
<!ENTITY ugrave  "&#249;"><!- small u, grave accent ->
<!ENTITY uacute  "&#250;"><!- small u, acute accent ->
<!ENTITY ucirc  "&#251;"><!- small u, circumflex accent ->
<!ENTITY uuml   "&#252;"><!- small u, dieresis or umlaut ->
<!ENTITY yacute "&#253;"><!- small y, acute accent ->
<!ENTITY thorn  "&#254;"><!- small thorn, Icelandic ->
<!ENTITY yuml   "&#255;"><!- small y, dieresis or umlaut ->
```

وبدلاً من تضمين تعليمات البرمجة ٩-١٨ في مجموعة DTD الفرعية الداخليـــة لمســتندك، يمكنك ببساطة استخدام مرجع كينونة معامل ليرتبط بها، ثم استخدم مرجع كينونـــة عامـــة فـــي مستندك.

فعلى سبيل المثال، بفرض أنك أردت وضع مستند Hildebrandslied، وهي إحدى القصائـد المعروفة في العصر الوسيط، بموقع الويب داخل مستند XML صحيح التكوين. إلا أنه بما أن هذا المخطوط مكتوباً بالألمانية، فهو يستخدم أحرف غير ASCII، ê وÎ وô وû وæ.

وللحصول على أعلى مستوى للقابلية للتحميل، يمكنك طباعة القصيـــدة فـــي ASCII، أثنـــاء تحويل هذه الحروف إلى تعليمـــات برمجيـــة كمرجـــع كينونـــة ê وî وô وû وæ على التوالي، إلا أنه حتى إذا لم تكن تطلب مستند صحيح متكامل، فستكون لازلت في حاجة إلى DTD لإعلان ذلك وأية مراجع كينونات أخرى قد تستخدمها. وأسهل الطرق للحصول على الأحرف الإضافية التي تحتاجها، فقط أن تشير إلـــى DTD الخـــارجي لتعليمـــات البرمجة ١٨-٩. وهذا ما توضحه تعليمات البرمجة ٩-١٩.

تعليمات البرمجة ٩-١٩: مستند صحيح التكوين غير صالح يستخدم مراجع كينونات للأحرف غير ASCII ISO-Latin-1

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY % ISOlat1
    PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN//XML"
        "http://www.schema.net/public-text/ISOlat1.pen">
    %ISOlat1;
]>
<DOCUMENT>
  <TITLE>Das Hildebrandslied, circa 775 C.E. </TITLE>
  <LINE>Ik gih&ocirc;rta dhat seggen,</LINE>
  <LINE>dhat sih urh&ecirc;ttun &aelig;non muot&icirc;n,</LINE>
  <LINE>Hiltibrant enti Hadhubrant untar heriun tu&ecirc;m.
  </LINE>
  <LINE>sunufatarungo: iro saro rihtun,</LINE>
```

```
  <COMMENT>I'll spare you the next 61 lines</COMMENT>
</DOCUMENT>
```

ويتكون الجزء الخاص بالمستند من XML صحيح التكوين يستخدم علامات تم إنشاؤها. وهـــي ليست معلنة في DTD ولا تتطلب أن تكون فقط لمستند صحيح التكوين إلا أن مراجع الكينونـــات تتطلب إعلانها في DTD إما في المجموعة الفرعية الداخلية أو الخارجية. وتقوم تعليمات البرمجة ٩-١٩ بإعلانها في المجموعة الفرعية الخارجية باستخدام مرجع كينونـــة المعـــامل الخارجيـــة %ISO latl لتحميل الكينونات المعلنة في تعليمات البرمجة ٩-١٨.

وتفيد DTDs أيضاً في تخزين النص المتداول الشائع المستخدم عبر موقع ويـــب لمســـتندات XML صحيحة التكوين، تماماً كما لو كانت مستندات XML صالحة. ويكون الإجراء أسهل بعـض الشيء عند العمل باستخدام مستندات XML صحيحة التكوين فقط حيث أنه لا توجد فرصة لأن لا يفي النص المتداول الذي تقوم بإدراجه بقيود DTD للمستند الأصل.

وفي البداية، قم بوضع النص المتداول داخل ملف بدون DTD، كما هو موضـــح بتعليمـــات البرمجة ٩-٢٠.

تعليمات البرمجة ٩-٢٠: نص إرشادات متداول بدون DTD

```
<?xml version="1.0"?>
<SIGNATURE>
  <COPYRIGHT>1999 Elliotte Rusty Harold</COPYRIGHT>
  <EMAIL>elharo@metalab.unc.edu</EMAIL>
</SIGNATURE>
```

وبعد ذلك، قم بكتابة DTD صغير كما في تعليمات البرمجة ٩-٢١ الذي يقوم بتعريف مرجـــع كينونة للملف الموجود في تعليمات البرمجة ٩-٢٠. وهنا، نفترض أنك قمـــت بتحديـــد موضـــع البرمجة ٩-٢٠ في دليل النص المتداول لملف Signature.xml، بالمســـتوى الجـــذري لخـــادم الويب، وأن بإمكانك البحث عن تعليمات البرمجة ٩-٢١ في ملف Signature.dtd فـــي دليـــل تعريفات نوع المستند بالمستوى الجذري لخادم الويب.

تعليمات البرمجة ٩-٢١: DTD للإرشادات، يقوم بتعريف مرجع كينونة

```
<!ENTITY SIGNATURE SYSTEM "/boilerplate/signature.xml">
```

ويمكنك الآن استيراد ملف Signature.dtd إلى أي مستند، ثم استخدام مرجع الكينونة العـام &SIGNATURE; لتضمين محتويات Signatur.xml بملفك. وهذا ما توضحه تعليمات البرمجـة ٩-٢٢.

تعليمات البرمجة ٩-٢٢: ملف يستخدم &SIGNATURE;

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY % SIG SYSTEM "/dtds/signature.dtd">
  %SIG;
]>
<DOCUMENT>
  <TITLE>A Very Boring Document</TITLE>
  &SIGNATURE;
</DOCUMENT>
```

وقد يبدو هذا مستوى أعلى للهدف، أكثر مما تحتاج بالفعل. فعلى سبيل المثال، تقوم تعليمـات البرمجة ٩-٢٣ بتعريف مرجع الكينونة &SIGNATURE; مباشرة في مجموعة DTD الفرعيـة الداخلية الخاصة به، ولقد أفاد ذلك بالفعل. إلا أن هذا المستوى الإضافي الأعلى للـهدف يوفـر حماية ضد إعادة تنظيم موقع الويب بينما لا تستطيع فقط تغيير الإرشـادات المسـتخدمة بكـل صفحات الويب الخاصة بك بتحرير ملف واحد. ومن جانب أخر فإن الطريقـة المباشـرة فـي تعليمات البرمجة ٩-٢٢ تسمح بصورة أكثر سهولة بإرشادات مختلفة في صفحات مختلفة.

تعليمات البرمجة ٩-٢٣: ملف يستخدم &SIGNATURE; مع مستوى أقل من الإمكانات

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ENTITY SIGNATURE SYSTEM "/boilerplate/signature.xml">
]>
<DOCUMENT>
  <TITLE>A Very Boring Document</TITLE>
  &SIGNATURE;
</DOCUMENT>
```

خلاصة

لقد اكتشفت من خلال هذا الفصل أن مستندات XML تم إنشاؤها من كل من الكينونات الخارجية والداخلية. وعلى وجه التحديد، لقد تعرفت في هذا الفصل على ما يلي:

- الكينونات هي وحدة التخزين الطبيعية التي يتم تجميع المستندات منها.
- تحتوي الكينونة على محتوى: XML صحيح التكوين وأشكال أخرى من النص وبيانات ثنائية.
- يتم تعريف الكينونات الداخلية كليةً داخل المستند، بينما ترسم الكينونات الخارجية محتواها من مصادر أخرى يتم تعيين موضعها عن طريق URL.
- تحتوي مراجع الكينونات العامة على الشكل &NAME; وتستخدم في محتوى المستند.
- يتم استبدال مراجع الكينونات الداخلية العامة بواسطة قيمة كينونة يتم تقديمها في إعلان الكينونة.
- يتم استبدال مراجع الكينونات الخارجية العامة بواسطة البيانات في URL والمحددة داخل إعلان الكينونة بعد كلمة System الأساسية.
- تحتوي مراجع كينونات المعاملات الداخلية على الشكل %NAME; وتستخدم بصفة خاصة في DTDs.
- يمكن دمج DTDs مختلفة بواسطة مراجع كينونات معاملات خارجية.
- تتيح مراجع الكينونات الخارجية إنشاء مستندات مركبة كبيرة من أجزاء صغيرة.

- هناك مستوى ثالث للمطابقة إلى معيار XML صحيح التكوين ولكن ليس صالحاً. وهـــذا إما بسبب عدم اكتمال DTD أو أن المستند لا يلتزم بقيود DTD.

وعند استخدام سمات المستند، لابد أن يتم أيضاً إعلان هذه الســـمات فـــي DTD. ويوضـــح الفصل التالي كيفية إعلان سمات في DTDs، وكيف يمكنك لذلك إرفاق قيود بقيم السمات.

١٠

إعلانات السمات في DTD

هناك سمات لبعض عناصر XML والتي تقدم المعلومات اللازمة لإجراء التطبيقات. والهدف من تلك السمات هو إعطاء معلومات إضافية خاصة بعنصر ما "مثل الرقم المعرف" يستخدمه فقط إحدى تلك البرامج التي تقرأ وتكتب الملف، وليست خاصة بمحتوى العنصر الذي يقرأه ويكتبه الفرد. في هذا الفصل ستتعرف على الأنواع المختلفة للسمات وكيفية إعلانها في DTD.

ما هي السمة

كما جاء في الفصل الثالث نرى أنه قد تحتوي العلامات البادئة والفارغة على أزواج من السمات الخاصة بالاسم والقيمة، يفصل بينهما علامة تساوي (=) مثال:

```
<GREETING LANGUAGE="english">
 Hello XML!
 <MOVIE SOURCE="WavingHand.mov"/>
</GREETING>
```

يتضح من المثال السابق أنه لعنصر Greeting سمة language التي لها قيمة English له سمة Source التي لها قيمة Waving Hand، ومحتوى عنصر Greeting هو Hello XML!. أما عن اللغة التي تكتب بها المحتوى فهي تقدم معلومات مفيدة عنه غير أنها لا تعد جزءاً منه.

وبالمطابقة، نجد أن محتوى عنصر Movie هو البيانات الثنائية المخزنة بالملف Waving Hand.Mov. واسم الملف لا يعد هو المحتوى على الرغم من أن الاسم هو الذي يحدد مكان المحتوى. لذا نقول مرة أخرى أن السمة تقدم معلومات عن محتوى العنصر لا عن المحتوى فحسب. وقد يكون للعناصر أكثر من سمة. مثال على ذلك:

```
<RECTANGLE WIDTH="30" HEIGHT="45"/>
<SCRIPT LANGUAGE="javascript" ENCODING="1-8859">...</SCRIPT>
```

في هذا المثال نجد أن سمة Language لعنصر Script لها قيمة Javascript. وسمة Encoding لعنصر Script لها قيمة 8859-1. وسمة Width لعنصر Rectangle لها قيمة 30 وسمة Height لعنصر Rec لها القيمة 45. وتعد كل هذه القيم سلاسلاً وليست أرقاماً.

لا يمكن أن يكون للعلامات الخاتمة سمات، فهذا المثال غير صحيح:

```
<SCRIPT>...</SCRIPT LANGUAGE="javascript" ENCODING="1-8559">
```

إعلان السمات في DTD

كما هو الحال بالنسبة للعناصر والكيانات، يجب إعلان السمات في DTD وذلك لضمان صلاحية المستند المستخدم فيه تلك السمات. وتوضح علامة <!Attlist> السمات ويكون لها هذا الشكل:

```
<!ATTLIST Element_name Attribute_name Type Default_value>
```

Element-name هو اسم العنصر الذي يشتمل على السمة وAttribule-name هــو اســم السمة. Type هو أحد الأنواع العشرة الصالحة الخاصة بالسمات والتي ذكرت في الجـدول ١٠-١. CDATA هو أكثر الأنواع شيوعاً وأخيراً Default-Value هو القيمة التي تتخذها السمة فــي حالة عدم تحديد قيمة معينة لها.

مثال على ذلك، هذا العنصر:

```
<GREETING LANGUAGE="Spanish">
  Hola!
</GREETING>
```

هذا العنصر يمكن إعلانه في DTD كالآتي:

```
<!ELEMENT GREETING (#PCDATA)>
<!ATTLIST GREETING LANGUAGE CDATA "English">
```

توضح علامة <!element> أن هناك عنصر تحية (greeting) يحتــوي علــى أحــرف موزعة. هذا ليس بالشيء الجديد. وعلامة <!ATTLIST> تدل على أن عناصر Greeting لــها سمة تحمل اسم Langage ذات قيمة من نوع CDATA التي هي بنفس قدر أهميتــه #Pcdata بالنسبة لمحتوى العنصر. وإذا حدث أن وجدت علامة Greeting بدون السمة Language، فإن القيمة English يتم استخدامها افتراضياً.

الجدول ١٠-١
أنواع السمات

النوع	المعنى
CDATA	بيانات أحرف/نص بدون علامات.
Enumerated	قائمة بالقيم الممكنة التي يتم اختيار واحدة منها.
ID	اسم فريد لا تستخدمه أية سمة أخرى من نوع الرقــم المعـرف فــي المستند.
IDREF	قيمة سمة الرقم المعرف لعنصر في المستند عدة قيم معرفــة لعنــاصر عدة يفصل بينها مسافة بيضاء.
Entity	اسم الكيان المعلن في DTD.
Entites	أسماء كيانات عدة معلنة في DTD يفصل بينها مسافة بيضاء.
Nmtoken	اسم من أسماء XML.

الجدول ١٠-١
أنواع السمات

النوع	المعنى
Notation	أسم ملاحظة معلنة في DTD.
Nmtokens	أسماء عديدة XML تفصل بينها مسافة بيضاء.

يتم إعلان قائمة السمات منفصلة عن العلامة نفسها، يكون اسم العنصر الذي تنتمي إليه السمة ضمن العلامة <!ATTLIST> يطبق إعلان السمة تلك على ذلك العنصر الذي هو GREEING في المثال السابق. في حالة وجود سمات LANGAGE لعناصر أخرى، فإن ذلك يتطلب إعلانات <!ATTLIST> منفصلة.

وكما هو الحال بالنسبة لمعظم الإعلانات، لا يعد الترتيب الدقيق الذي يظهر عليه إعلانات السمات مهماً، فقد يأتوا قبل أو بعد إعلان العنصر التابع لهم. في حقيقة الأمر، يمكنك إعلان سمة أكثر من مرة "بالرغم من أنني لا أنصح بذلك" يأخذ فيها الإعلان الأول الأولوية.

بل ويمكنك أيضاً إعلان سمات لعلامات لا يوجد لها، على الرغم من عدم شيوع تلك الظاهرة. قد تقوم أحياناً بإعلان تلك السمات غير الموجودة كجزء من التحرير المبدئي لـ DTD، بخطة العودة إليها مرة أخرى فيما بعد وإعلان العناصر.

إعلان السمات المتعددة

عادة ما يكون للعناصر سمات متعددة، فعنصر IMG في HTML يمكن أن يكون له HEIGHT، WIDTH، ALT، BORDER، ALIGN إلى غير ذلك من السمات. وفي حقيقة الأمر فإن معظم العلامات الخاصة بـ HTML قد يكون لها سمات متعددة، كذلك الحال بالنسبة لعلامات XML. فعلى سبيل المثال عنصر مثل RECTANGLE يحتاج إلى LENGTH وWIDTH.

```
<RECTANGLE LENGTH=٧٠"px" WIDTH="85px"/>
```

يمكنك إعلان هذه السمات بأشكال مختلفة، متضمنة إعلاناً واحداً لكل سمة، مثال:

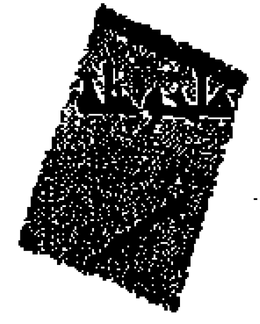
```
<!ELEMENT RECTANGLE EMPTY>
<!ATTLIST RECTANGLE LENGTH CDATA ٠"px">
<!ATTLIST RECTANGLE WIDTH  CDATA ٠"px">
```

ومن ثم يوضح المثال السابق أن عناصر RECTANGLE تتضمن سمات LENGTH وWIDTH، كل منهما يحمل القيمة الافتراضية OPX وبإمكانك أيضاً دمج علامتي <!ATTLIST> في إعلان واحد كالتالي:

```
<!ATTLIST RECTANGLE LENGTH CDATA ."px"
                   WIDTH  CDATA ."px">
```

يوضح الإعلان الواحد سمتي LENGTH وWIDTH، كل منها من نــــوع CDATA، ويتخــذ القيمة الافتراضية OPX. باستطاعتك أيضاً استخدام هذا البناء للجملة عندما يكون للسمات أنواعــاً مختلفة كما هو موضح:

```
<!ATTLIST RECTANGLE LENGTH CDATA "15px"
                   WIDTH  CDATA "34px"
```



أنا شخصياً لست مغرماً بهذا النمط، فهو يبدو مشوشاً ويعتمد اعتماداً كبيراً علــــى إدراج مسافات بيضاء حتى يكون مقروءاً وواضحاً على الرغـــــم مـــن أن تلـــك المسافات ليست بهذا القدر من الأهمية بالنسبة للمعنى الأساسي للعلامة، غير أنــك ستتعرض لهذا النمط في DTD الذي يكتبه آخرون لذا كان عليك معرفته.

تحديد القيم الافتراضية للسمات

بدلاً من تحديد قيمة سمة افتراضية واضحة مثل OPX يمكن لإعلان سمة أن يتطلب من الكـــاتب أن يقوم بتزويد قيمته، كما يسمح أيضاً بتجاهل القيمة تماماً، أو حتى بالاســـتخدام الدائـــم للقيمـــة الافتراضية. وقد حددت تلك المتطلبات بثلاث كلمات أساسية هــــي بـــالترتيب: #REQUIRED و#IMPLIE و#FIXED.

#REQUIRED

قد لا يكون أمامك خيار جيد لقيمة افتراضية ما. فعلى سبيل المثال عند كتابة DTD لاســـتخدامه على الإنترنت، يتطلب ذلك أن تحتوي جميع المستندات على علامة <Author/> واحدة فارغـــة على الأقل. هذه العلامة بطبيعة الحال لا يمكن أداؤها لكنها يمكن أن تحدد الشخص الـــذي قـــام بإنشاء المستند وقد تحتوي هذه العلامة على سمات عـــدة مثـــل Name، Email وExtension فيمكن بذلك الاتصال بالكاتب: مثال:

```
<AUTHOR NAME="Elliotte Rusty Harold"
  EMAIL="elharo@metalab.unc.edu" EXTENSION="3459"/>
```

وبدلاً من تزويد قيم افتراضية لتلك السمات، افترض أنك تريد أن تخبر شخصاً ما على توزيع مستند على الإنترنت لتحديد هويته. فبينما لا يمكن لأي XML أن تمنع شـــخصاً مـــن أن ينســـب التأليف لشخص ما باستخدام #REQUIRED كقيمة افتراضية مثال:

```
<!ELEMENT AUTHOR EMPTY>
<!ATTLIST AUTHOR NAME CDATA #REQUIRED>
<!ATTLIST AUTHOR EMAIL CDATA #REQUIRED>
<!ATTLIST AUTHOR EXTENSION CDATA #REQUIRED>
```

وإذا حدث أن واجه الموزع علامة <author/> لا تحتوي على واحدة أو أكثر من تلك السمات، فإنه سيكون هناك خطأ ما.

قد تحتاج أيضاً إلى استخدام #REQUIRED لإجبار المؤلفين على أن يعطوا عناصر IMG، Width وHeight وسمات Alt، مثال:

```
<!ELEMENT IMG EMPTY>
<!ATTLIST IMG ALT    CDATA #REQUIRED>
<!ATTLIST IMG WIDTH  CDATA #REQUIRED>
<!ATTLIST IMG HEIGHT CDATA #REQUIRED>
```

أي محاولة لتجاهل تلك السمات "كما يحدث في كثير من صفحات ويب" يجعل المستند غير صالح. وعلى الفور، يلحظ معالج XML الخطأ ويخطر المؤلف بالسمات الناقصة.

#IMPLIE

في بعض الأحيان قد لا يكون أمامك خيار جيد لقيمة افتراضية، وفي نفس الوقت لا تريد من مؤلف المستند أن يضع قيمته. فلنفترض مثلاً أن بعضاً ممن يقومون بتوزيع المستندات على الإنترنت بك يعملون خارج الموقع على مسئوليتهم دون الارتباط بشركة، ولديهم عناوين البريد الإلكتروني، لكن ليس لديهم تليفون موصل بالخط الأساسي "امتدادي"، لذا لا يتعين عليك مطالبتهم باستخدام سمة امتدادية في علامات <author/> الخاصة بهم. مثال:

```
<AUTHOR NAME="Elliotte Rusty Harold"
      EMAIL="elharo@metalab.unc.edu" />
```

أنت لا زلت لا ترغب في وضع قيمة افتراضية للامتداد، لكنك ترغب في السماح للمؤلف بوضع سمة من هذا النوع. في هذه الحالة استخدم #IMPLIED كقيمة افتراضية كالتالي:

```
<!ELEMENT AUTHOR EMPTY>
<!ATTLIST AUTHOR NAME      CDATA #REQUIRED>
<!ATTLIST AUTHOR EMAIL     CDATA #REQUIRED>
<!ATTLIST AUTHOR EXTENSION CDATA #IMPLIED>
```

وإذا حدث أن واجه موزع XML علامة <author/> بدون سمة EXTENSION يتم إبــــلاغ تطبيق XML بأنه لا تتوافر أي قيمة. في هذه الحالة يقوم التطبيق بالعمل طبقاً لذلك كمـــــا يقـــوم بعملية الاختيار. على سبيل المثال إذا كان التطبيق هو تغذية قــــاعدة بيانـــات SQL بمجموعـــة عناصر، حيث تعين السمات في حقول فإنه من المحتمل أن يقوم بإدراج قيمة خالية فـــــي حقــل قاعدة البيانات المتماثل.

#FIXED

وأخيراً قد ترغب في تزويد قيمة افتراضية للسمة دون السماح للمؤلف بتغييرها. فإنك قد تتمنـــــى مثلاً أن تقوم بتحديد سمة Company مطابقة لعنصــــــر Author لأي شـــخص يقـــوم بتوزيـــع مستندات على الإنترنت كالأتي:

```
<AUTHOR NAME="Elliotte Rusty Harold" COMPANY="TIC"
  EMAIL="elharo@metalab.unc.edu" EXTENSION="3459"/>
```

ويمكن أن تطلب أن يستخدم الجميع هذه القيمة الخاصة بالشركة (company) وذلك بإعطــــاء القيمة الافتراضية اسم fixed# ثم الإتيان بعد ذلك بالافتراض الأساسي، مثال:

```
<!ELEMENT AUTHOR EMPTY>
<!ATTLIST AUTHOR NAME      CDATA #REQUIRED>
<!ATTLIST AUTHOR EMAIL     CDATA #REQUIRED>
<!ATTLIST AUTHOR EXTENSION CDATA #IMPLIED>
<!ATTLIST AUTHOR COMPANY   CDATA #FIXED "TIC">
```

ومؤلفا المستند غير مطالبين بتضمين السمة الثابتة في علاماتهم، وفي حالة عدم تضمينها فإنه يتم استخدام القيمة الافتراضية، أما في حالة تضمينها فإنه يتعين عليهم استخدام قيمة مطابقة، وإلا فإنه سيكون هناك خطأ ما.

أنواع السمات

تحتوي كل الأمثلة السابقة على سمات من نوع CDATA الذي يعد النوع الأكثر شيوعاً، غير أنــــه هناك أيضاً تسع أنواع أخرى وهذه الأنواع العشر هي:

- CDATA
- Enumerated
- NMTOKEN

- ♦ NMTOKENS
- ♦ ID
- ♦ IDREF
- ♦ IDREFS
- ♦ ENTITY
- ♦ ENTITIES
- ♦ NOTATION

تسع من هذه السمات تعد ثوابت تستخدم في حقل النوع بينما Enumerated نوعــاً متمــيزاً يوضح أن السمة يجب أن تتخذ قيمتها من قائمة لكل القيم الممكنة. والآن لنتعرف على كل نـــوع على حدة بعمق أكثر.

نوع سمة CDATA

يقصد بالـــ CDATA، والتي تعد من أكثر الأنواع شيوعاً، أن تكون قيمة السمة أي نص سلســـلة يرد فيه علامة أقل من (<) أو علامة التنصيص (") وغير أن هذه الأحـــرف يمكـــن إدراجـــها باستخدام مراجع الكيانات المعتادة "< و"" أو باستخدام قيم اليونيكـــود الخاصـــة بتلـــك الأحرف عن طريق مراجع الأحرف. إضافة إلى ذلك نجد أن كل علامات الضم (&) أي علامـــة الضم التي لا تبدأ حرف أو مرجع كيان، يجب استخدام &amp للخروج منها.

حتى في حالة احتواء القيمة ذاتها على علامات اقتباس مزدوجة، فذلـــك لا يســـتدعى منـــك الهروب، بل يمكنك بدلاً من ذلك لاستخدام علامات اقتباس مفردة لتحديد السمات كما في المثـــال التالي:

<RECTANGLE LENGTH='7"' WIDTH="8.5""/>

أما في حالة احتواء قيمة السمة على علامات اقتباس مفردة ومزدوجة معاً فإنه يتم اســـتبدال العلامات غير المستخدمة في تحديد القيمة بمراجع الكيـــان: &apos "فاصلـــة عليـــا" و&quot "علامة اقتباس مزدوجة" مثال:

<RECTANGLE LENGTH='8'7"' WIDTH="10'6""/>

نوع سمة ENUMERATED

لا يعد هذا النوع من الكلمات الأساسية لـــ XML، بل هو قائمة بالقيم الممكنة للسمة، يفصل بينهما أشرطة عمودية، ويجب أن تكون كل قيمة اسم XML صالح. يمكن لمؤلف المستند أن يختار أيـــاً من هذه القيم فتكون بذلك هي القيمة الافتراضية.

مثال: فلنفترض أنك تريد أن تجعل عنصراً ما مرئياً أو غير مرئي، فإنك قد تحتاج لأن تجعل سمة Visible للعنصر التي يكون لها فقط قيمتي TRUE أو FALSE. إذا كان العنصر هو عنصر P البسيط، فسيكون شكل الإعلان كالآتي:

```
<!ATTLIST P VISIBLE (TRUE | FALSE) "TRUE">
```

يوضح هذا الإعلان السابق أن عنصر P قد يكون أولا يكون له سمة Visible. في حالة وجود سمة Visible، ستكون قيمة السمة حتماً TRUE أو FALSE، أما في حالة عدم وجودها فستكون القيمة TRUE، مثال:

```
<P VISIBLE="FALSE">You can't see me! Nyah! Nyah!</P>
<P VISIBLE="TRUE">You can see me.</P>
<P>You can see me too.</P>
```

لكن ليس هذا الإعلان بتعويذة سحرية تمكنك من إخفاء النص، بل هو يعتمد على التطبيق حتى لا يقوم بعرض عناصر غير مرئية. وسواء كان العنصر ظاهر أم خفي، سيتم إعداده من خلال قاعدة نمط الورق تطبق على العناصر ذوات السمات Visible، مثال:

```
<xsl:template match="P[@VISIBLE='FALSE']">
</xsl:template>
<xsl:template match="P[@VISIBLE='TRUE']">
  <xsl:apply-templates/>
</xsl:template>
```

The NMTOKEN Attribute Type

نوع سمة NMTOKENS

يقصر هذا النوع قيمة السمة على اسم XML صالح، وكما ذكر سلفاً في الفصل السادس، يجب أن تبدأ أسماء XML بحرف أو بتسطير أسفل السطر (_) كما أن هناك أحرف تالية في الاسم قد تتضمن أحرف، أعداد وتسطير أسفل السطور، واصلات ونقاط، ولكنها لا تتضمن مسافات بيضاء (عادة ما يحل التسطير أسفل السطر محل المسافات البيضاء). ومن الناحية الفنية قد تتضمن تلك الأسماء علامة النقطتين لكن عليك ألا تستخدمها لأنها مخصصة للاستخدام مع مسافات الأسماء.

وتظهر أهمية Nmtoken عند استخدام لغة برمجة لمعالجة بيانات XML. فهو ليس من قبيل الصدفة باستثناء علامات النقطتين المسموح بها أن تتطابق هذه القواعد السابقة مع قواعد المعرفات في جافا (Javo) وجافا سكريبت (Javascript) وغيرهما من لغات البرمجة. فعلى

سبيل المثال يمكنك استخدام Nmtoken لجعل فئة Java معينة مقترنة بعنصر ما. ومن ثم يمكنك استخدام انعكاس AP1 لجافا (Java) لتمرير البيانات إلى طريقة معينة داخل فئة معينة.

كما يساعدك أيضاً هذا النوع من السمات على اختيار اسم ما من مجموعة من الأسماء ليست جزءاً من XML، لكنها لها بعض مواصفات اسم XML، أبرزها منع استخدام المسافات البيضاء. مثال يمكنك استخدام Nmtoken مع سمة يكون قيمتها لملف يحمل اسم 8.3 DOS، لكن لا يمكن استخدامها مع Macintosh-Unix أو مع اسم الملف Windows NT، لأن تلك الأسماء عادة ما يستخدم معها مسافات بيضاء، مثلاً قد تود اختصار سمة لولاية ما في علامة <address/> إلى حرفين. إنك لن تستطيع استخدام تلك الخاصية مع DTD، لكن باستطاعتك منع الناس من دخول "New york" أو "Purto Rico" وذلك عن طريق إعلان <ATTIST!> التالي:

```
<!ATTLIST ADDRESS STATE NMTOKEN #REQUIRED>
```

غير أن بعض الولايات مثل كاليفورنيا "California" و نيفادا "Nevada" وغيره ممن تتكون أسماؤها من كلمات مفردة ما زالت تمثل قيماً قانونية. بالطبع يمكنك الاستعانة بقائمة مرقمة لعشرات التعليمات البرمجية المكونة من حرفين لكن اتباع هذا النهج سوف يستغرق ساعات عمل أكثر. فهل ستكون مثلاً على دراية بكل التعليمات البرمجية المكونة من حرفين للولايات الأمريكية البالغ عددها DTD، فيمكنك إعادة استخدام الملف مرات عديدة.

نوع سمة NMTOKENS

هي صيغة الجمع لسمة Nmtoken لكنها نادرة الاستخدام بواسطتها يمكن أن تتكون قيمة السمة من أسماء XML متعددة، يفصل بينها مسافات بيضاء. عامة لـ Nmtokens نفس استخدامات Nmtokens، لكنه يستخدم فقط في حالة وجود أسماء متعددة.

فإذا رغبت مثلاً في معرفة تعليمات برمجية متعددة عن ولاية مكونة من حرفين عن سمة الولاية، انظر المثال التالي:

```
<!ATTLIST ADDRESS STATES NMTOKENS #REQUIRED>
```

ثم سيظهر لك علامة عنوان (address) كالتالي:

```
<ADDRESS STATES="MI NY LA CA">
```

لكن من ضمن العيوب أنك لو استعنت بتلك التقنية فلن يكون هناك مجالا للبحث في بعض الولايات مثل نيويورك حيث أن كل فرد يعد جزء من اسم الولاية يحدد Nmtoken كالآتي:

```
<ADDRESS STATES="MI New York LA CA">
```

نوع سمة ID

ينفرد هذا النوع عن غيرة بقدرته على تحديد العنصر في المستند لذلك المستند دون الاهتمام بمعرفة معانيهم أو بعلاقة كل منهم بالآخر

ويجب أن تكون قيمة السمة من نوع ID أسم XML صالح أي أن يبدأ بحرف وأن يتكون من الحروف الهجائية العددية وتسيطر أسفل السطر دون أن يكون هناك مسافات بيضاء كما لا يمكن استخدام الاسم كسمة ID لأكثر من علامة فاستخدم نفس الرقم المعروف لأكثر من مرة ينشب عنة خطاء ما إضافة إلى ذلك لا يكون لكل عنصر أكثر من سمة واحدة من نوع ID.

وبالمثل تتواجد سمات ID منفردة لتلائم البرامج التي تعالج البيانات وفي كثير من الحالات قد يكون هناك عناصر متعددة متطابقة في كل شي فيما عدا قيمة سمة ID إذا اخترت بعضا من الأرقام المعرفة في نمط قابل لأن يتنبأ به سيقوم برنامج يعد كل العناصر المختلفة للأنواع المختلفة أو لنوع واحد داخل المستند.

وهذا النوع يعد غير متوافق مع Flexed لذا لا نجد سمة ثابتة (Fined) وفي نفس الوقت من نوع ID لأن سمة Fixed يكون لها واحدة بينما لكل سمة من نوع ID قيمة مختلفة تبعا لذلك تستخدم معظم سمات Required ID كما توضح تعليمات البرمجة ١٠-١.

تعليمات البرمجة ١٠-١: سمة ID مطلوبة

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT (P*)>
  <!ELEMENT P (#PCDATA)>
  <!ATTLIST P PNUMBER ID #REQUIRED>
]>

<DOCUMENT>
  <P PNUMBER="p1">The quick brown fox</P>
  <P PNUMBER="p2">The quick brown fox</P>
</DOCUMENT>
```

The IDREF Attribute Type

نوع سمة IDREF

لكون قيمة السمة من نوع IDREF هي الرقم المعرف لعنصر أخر في المستند مثال علـــى ذلـــك تعليمات البرمجة ١٠-٢ والتي توضح سمات IDREF وID التي تســـــتخدم لتوصيـــل الأطفـــال (Chides) بذويهم (Parents).

تعليمات البرمجة ١٠-٢ Family.XML

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT (PERSON*)>
  <!ELEMENT PERSON (#PCDATA)>
  <!ATTLIST PERSON PNUMBER ID #REQUIRED>
  <!ATTLIST PERSON FATHER IDREF #IMPLIED>
  <!ATTLIST PERSON MOTHER IDREF #IMPLIED>
]>

<DOCUMENT>
 <PERSON PNUMBER="a1">Susan</PERSON>
 <PERSON PNUMBER="a2">Jack</PERSON>
 <PERSON PNUMBER="a3" MOTHER="a1"
FATHER="a2">Chelsea</PERSON>
 <PERSON PNUMBER="a4" MOTHER="a1"
FATHER="a2">David</PERSON>
</DOCUMENT>
```

يكثر استخدام هذا النوع الدقيق، على الرغم من عدم شيوعه، لإقامة صلات بين العناصر غير الموجودة في البناء المتفرع للمستند. ففي تعليمات البرمجة ١٠-٢، يحصل كـــل طفـــل (Child) على سمتي Father وMother والتي تحتوي على سمات ID الخاصة بالأب والأم.

غير أنه ليس من السهل استخدام Idref مباشرة لربط الآباء بأبنائهم في تعليمـــــات البرمجـــة ١٠-٢، وذلك لأن كلاً من الآباء له عدد غير محدد من الأبناء، لكنك على أية حال يمكنك جمـــع كل الأبناء ممن لهم نفس الآباء في عنصر Family وربطه بـــ Family. إلا أن هذا النهج أيضـــاً لا يمكن استخدامه مع الأبناء غير الأشقاء. لذا باختصار يمكن القول بأن Idref يمكن اســـتخدامه لتحديد العلاقات وانتقالها من كثيرين إلى واحد فقط وليس العكس.

نوع سمة ENTITY

يمكنك هذا النوع من ربط البيانات المزدوجة الخارجية، أي ربط الكيان العام الخارجي غير المحلل بالمستند، وقيمة هذه السمة تكون اسماً لكيان عام غير محلل معلن في DTD وتكون مرتبطة بالبيانات الخارجية.

والمثال الكلاسيكي لسمة Entity هو الصورة (Image) التي تتكون من معلومات مزدوجة مأخوذة من URL آخر، وإذا لم يدعمها عارض XML، يمكنك إدخال صورة أخرى في مستند XML في DTD بالإعلانات التالية:

```
<!ELEMENT IMAGE EMPTY>
<!ATTLIST IMAGE SOURCE ENTITY #REQUIRED>
<!ENTITY LOGO SYSTEM "logo.gif">
```

ثم قم بإدراج علامة Image التالية في مكان الصورة المراد بالمستند:

```
<IMAGE SOURCE="LOGO"/>
```

لا تعد هذه الطريقة وصفة سحرية يفهمها عارض XML آلياً لكنها ببساطة إحدى تقنيات الاستعراض والتطبيقات الأخرى من الممكن أن تتخذ أولاً تتخذ بيانات ليست من نوع XML وتضمنها في المستندات.



ستعرض هذه التقنية بشكل أوضح في الفصل الحادي عشر "تضمين بيانات ليست من نوع XML".

سمة Entities

هذا النوع هو صيغة الجمع، النادرة نسبياً، للنوع Entity، وتتكون قيمة هذا النوع من أسماء كيانات متعددة غير محللة يفصل بينها مسافات بيضاء ويشير كل أسم كيان إلى مصدر خارجي لبيانات ليست من نوع XML. إحدى استخدامات هذه الطريقة هي عرض الشرائح على الشاشة التي تقوم بدورها بعرض الصور المختلفة كما هو مبين في المثال التالي:

```
<!ELEMENT SLIDESHOW EMPTY>
<!ATTLIST SLIDESHOW SOURCES ENTITIES #REQUIRED>
<!ENTITY PIC1 SYSTEM "cat.gif">
<!ENTITY PIC2 SYSTEM "dog.gif">
<!ENTITY PIC3 SYSTEM "cow.gif">
```

ثم قم بإدراج العلامة التالية في المكان الذي تريد فيه أن تظهر عرض الشرائح في المستند:

```
<SLIDESHOW SOURCES="PIC1 PIC2 PIC3">
```

مرة أخرى نقول أن هذه ليست الصيغة العامة التي بفهمها عارض XML أتوماتيكيـــاً، لكنـــها ببساطة طريقة يمكن لتطبيقات أخرى أن تقوم أولاً تقوم بتضمينها في بيانات ليست من نوع XML داخل المستندات.

نوع سمة NOTATION

يقضي هذا النوع بأن تكون قيمة السمة هي اسم لملاحظة معلنة في DTD، كما يجب أن تكـــون أيضاً القيمة الافتراضية لهذه السمة اسماً لملاحظة معلنة في DTD، وسنتعرض بتفصيـــل أكـــثر للملاحظات في الفصل التالي. لكن باختصار، يمكن القول بأن الملاحظات تنسق البيانات التي هي ليست من نوع XML، وذلك مثلاً عن طريق تحديد تطبيق لكيان غير محلل.

المرجع الفصل الحادي عشر "تضمين بيانات ليست من نوع XML" تقوم بتغطيـــة أكـــبر للملاحظات.

مثال: سمة Player لعنصر Sound هي من نوع Notation والقيمة الافتراضية هـــي MP، والملاحظة هنا تشير إلى نوع معين من ملف الصوت (Sound):

```
<!ATTLIST SOUND PLAYER NOTATION (MP) #REQUIRED>
<!NOTATION MP SYSTEM "mplay32.exe">
```

كما أنة يمكنك الاختيار بين الملاحظات المختلفة وذلك يمكنك من تحديد التطبيقات المســـاعدة المختلفة للنظم الأساسية المختلفة ثم يقوم العارض بانتقاء التطبيق المتوفر لدية في هــــذه الحالـــة تلحق الكلمة الأساسية Notation بمجموعة أقواس تحتوى على قائمة بأسماء الملاحظات الممكنـة يفصل بينها أشرطة عمودية مثال هذا يعنى أن سمة Player لعنصر Sound قد تعين في MPأو ST أوSM وسنطرق لهذا الموضوع بوضوح أكثر في الفصل اللاحق.

```
<!NOTATION MP SYSTEM "mplay32.exe">
<!NOTATION ST SYSTEM "soundtool">
<!NOTATION SM SYSTEM "Sound Machine">
<!ATTLIST SOUND PLAYER NOTATION (MP | SM | ST) #REQUIRED>
```



قد تبدو هذه الطريقة منذ الوهلة الأولى متعارضة مع التناول لقوائم سمات أخرى مثل Entities وnmtokens لكن هذان النوعان مختلفان إلى حد ما حيث لهما قائمة بالسمات في العنصر الأساسي في المستند لكن قيمة واحدة فقط في إعلان السمة في DTD. أما Notation فلها قيمة واحدة فقط في السمة الخاصة بالعنصر الأساسي في المستند، بينما توجد قائمة بجميع القيم الممكنة في إعلان السمة في DTD.

السمات المعرفة من قبل

هناك سمتان معرفتان من قبل في XML يجب إعلانها في DTD لكل عنصر يطبقان عليه، لكن يجب استخدامها في الأغراض المحدد لها القيام بها. وتحدد هاتان السمتان باسم تكون بدايته من نوع XML. هاتان السمتان هما xml: Space، وxml: lang تشرح xml: Space كيفية معالجة المسافات البيضاء في العنصر بينما توضح xml:lang لغة العنصر "وإذا أردت أيضاً اللهجة والبلد".

XML: Space

في HTML ليس هناك أهمية للمسافات البيضاء وعلى الرغم من أن الفرق بين وجود وعدم وجود مسافة يعد ملحوظا بدرجة كبيرة فان الفرق بين وجود مسافة واحدة ومسافتين أو مسافة واحدة وحرف إرجاع أو مسافة واحدة وثلاثة أحرف إرجاع وأثنتا عشر علامة جدول لا يكاد ملحوظا أما بالنسبة للنص الذي يكون فيه المسافات البيضاء مهمة مثل البرنامج المصدر للكمبيوتر وبعض تقارير قواعد البيانات فانه يمكنك استخدام عنصر PRE لتحديد الخط ذو المسافات الأحادية وللمحافظة على المسافات البيضاء.

غير أن XML يحافظ على المسافات البيضاء بطريقة افتراضية حيث يقوم معالج XML بنقل كل أحرف المسافات البيضاء إلى التطبيق دون أحداث أدنى تغيير بها وعادة ما يتجاهل التطبيق المسافات البيضاء الزائدة لكن في نفس الوقت يقوم معالج XML بأعلام التطبيق بوجود بعض العناصر التي تحتوى على مسافات بيضاء هامة يجب الاحتفاظ بها ولبيان تلك العناصر للتطبيق يستخدم مؤلف الصفحة سمة XML: Space.

وعندما يحتوى العنصر على مسافات بيضاء هامة، يجب أن يكون DTD علامة <!ATTLIST> لسمة XML :SPACE وهذه السمة سيكون بها نوع ENUMER ARED يحمل القيمتين DEFAULT وPRESERVE كما هو مبين في تعليمات البرمجة ٣-١٠.

تعليمات البرمجة ١٠-٣: البرنامج المصدر جافا (JAVO) والمسافات البيضاء الهامة رموز في XML

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE PROGRAM [
  <!ELEMENT PROGRAM (#PCDATA)>
  <!ATTLIST PROGRAM xml:space (default|preserve) 'preserve'>
]>
<PROGRAM xml:space="preserve">public class AsciiTable {
  public static void main (String[] args) {
for (int i = 0; i &lt; 128; i++) {
     System.out.println(i + "   " + (char) i);
   }
}
}
</PROGRAM>
```

يتم نقل كل المسافات البيضاء إلى التطبيق لغض النظر عما إذا كانت قيمةDEFAULT XML SOACE أوPRESERVE فمع القيمة DEFCUELT يتصرف التطبيق تماما مثلما يتصـرف مـع المسافات البيضاء الزائدة أما مع القيمة PRESERUE فان التطبيق يتعامل مع المسافات البيضـاء الزائدة باعتبارها ذات أهمية.



تعتمد درجة الأهمية إلى حد ما على الوجهة النهائية للبيانات فعلى سبيل المثـال تعد المسافات البيضاء الزائدة في برنامج المصدر جافا (JAVA) مرتبطة بمحـرر برنامج المصدر وليس بالمترجم.

وعن نتائج العنصر ذو السمة SPACE وXML فإنه يفترض تصرفهم تماما مثـل منتجاتـهم "سواء بالمحافظة أو عدم المحافظة على المسافة" إلا إذا كان عندهم سـمة SPACE أو XML ذات قيمة متعارضة.

XML: LANG

تحدد هذه السمة اللغة التي يكتب بها العنصر وتكون قيمتها من نوع CDATA أو NMTOKEN أو قائمة عديدة (ENUMER ARED LIST) لكنة يعد مثاليا أن تكون قيم تلك السمات إحدى التعليمات البرمجية للغة المكونة من حرفين والتي يعرفها قياس ISO-٦٣٩ الأصلي.

http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt

مثال انظر إلي هذه الجملة المأخوذة من SARLICON لبيترونيس كتبت مرة باللاتينية ومرة أخرى بالإنجليزية وعلى الرغم من وجود علامة الجملة في كلتا الجملتين فإن علامة الجملة الأولى لها سمة XML : LANG الخاصة باللاتينية بينما لعلامة الجملة الثانية نفس السمة لكن خاصة بالإنجليزية.

اللاتينية:

```
<SENTENCE xml:lang="la">
  Veniebamus in forum deficiente now die, in quo notavimus
  frequentiam rerum venalium, non quidem pretiosarum sed tamen
  quarum fidem male ambulantem obscuritas temporis
  facillime tegeret.
</SENTENCE>
```

الإنجليزية:

```
<SENTENCE xml:lang="en">
  We have come to the marketplace now when the day is failing,
  where we have seen many things for sale, not for the
  valuable goods but rather that the darkness of
  the time may most easily conceal their shoddiness.
</SENTENCE>
```

يستطيع من يتحدث ويقرأ الإنجليزية أن يحدد بسهولة ويسر النص الأصلي من المترجم، والكمبيوتر كذلك لكن فقط بملاحظة سمة &ml: Lang، وتمكن هذه القدرة على التفرقة مدقق الإملاء من التحقق من عنصر ما يشير إلى أي قاموس يستخدم. ويمكن لمحركات البحث أن تعاين سمات اللغة تلك لتحدد ما إذا كانت ستقوم بفهرسة الصفحة وإعادة المطابقات المبنية أساس على تفصيلات المستخدم.

لغات متعددة وتعليمات برمجية غير كافية

يبدو XML قليلا غير مناسب لوقتنا هذا، فإن قياس 639- ISO الأصلي للتعليمات البرمجية للغة يتكون من اثنين من الحروف الأبجدية لـ ASCII. هذا القياس لا يسمح بوجود أكثر من ٢٦ لا ٢٦ أو ٦٧٦ تعليمية برمجية، لكن في يومنا هذا يوجد أكثر من ٦٧٦ لغة مختلفة دون احتساب اللغات الميتة مثل اللغة الاتروزية وهي لغة بلاد قديمة في غرب إيطاليا. وعند التطبيق نجد أن التعليمات البرمجية لم يصل عددها حتى إلى ستمائة بستة وسبعين، وذلك لأن اختصارات اللغات يجب أن يكون شكلها إلى حد ما قريبا من اسم اللغة.

يستخدم الجزء الثاني في ISO-693 التعليمات البرمجية للغة المكونة من ثلاثة أحرف والتي ستتناول جميع اللغات الموجودة على الأرض، إلا أن قياس XML يتطلب تعليمات برمجية مكونة من حرفين على وجه الخصوص.

تطبيق اللغة على عنصر ما وعلى نتائج هذا العنصر، إلى أن تعلن إحدى تلك النتائج لغة جديدة، ويكون إعلان عنصر Sentence كالآتي:

```
<!ELEMENT SENTENCE (#PCDATA)>
<!ATTLIST SENTENCE xml:lang NMTOKEN "en">
```

في حالة عدم توافر تعليمية برمجية ISO مناسبة، يمكنك استخدام إحدى التعليمات المسجلة مع Iana، على الرغم من إنها عادة ما تستخدم ثلاث تعليمات برمجية إضافية "مذكورة في الجدول ١٠-٢". يمكنك أن تجد القائمة الأكثر شيوعاً على الموقع التالي:

http://www.isi.edu/in-notes/iana/assignments/languages/tags

الجدول ١٠-٢

تعليمات IANA البرمجية الخاصة باللغة

التعليمات البرمجية	اللغة
No-bok	Norwegian "Book language"
No-nyn	Norwegian "New Norwegian"
i-navajo	Navajo
i-mingo	Mingo

على سبيل المثال:

<P xml:lang="no-nyn">

لو أن أياً من ISO أو IANA ليس لهما تعليمية برمجية للغة التي تريدهـــا "كلغـــة كلينجـــون Klingon مثلاً" يمكنك استخدام تعليمات برمجية جديدة للغة، على أن تبدأ تلك التعليمات بالسلسلة -x أو -X حتى يتم التعارف عليها كتعليمات مستخدم معرفة، للاســتخدام للأغــراض الخاصـــة، مثال:

<P xml:lang="x-klingon">

قد تتضمن قيمة سمة XML:Lang أقساماً لتعليمات برمجية إضافية، يفصلها عــن التعليمـــات البرمجية الأساسية للغة واصلة. وفي أغلب الأحيان يكون القسم الأول لتعليمات البرمجة الفرعيـة تعليمية برمجية لبلدها مكونة من حرفين، يحددها ISO 3166، ومن ثم يمكنـــك أحيـاء قائمـة التعليمات البرمجية للبلاد الأكثر شيوعاً عن طريق:

http://www.isi.edu/in-notes/iana/assignments/country-codes.

مثال:

<P xml:lang="en-US">Put the body in the trunk of the car.</P>
<P xml:lang="en-GB">Put the body in the boot of the car.</P>

لو أن القسم الأول للتعليمية البرمجية الفرعية لا يمثل تعليمية البلاد البرمجية ISO المكونـــة من حرفين، يجب أن يكون تعليمية برمجية لمجموعة حروف للغة المســجلة مــع IANA مثــل csDECMCS، ROMAN 8 وmac وcp037 ebcdic-cp-cag وتوجد القائمة المتداولـــة علـــى ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets.

مثال:

<P xml:lang="en-mac">

والاحتمال الأخير هو أن تكون التعليمية البرمجية الفرعية تعليمية برمجية أخرى تبدأ بـــ -x أو -X، مثال:

<P xml:lang="en-x-tic">

وبحسب الاتفاق أصبحت التعليمات البرمجية للغة تكتب بأحرف صغيرة، بينما تكتـــب تلـك الخاصة بالبلاد بأحرف كبيرة، غير أن هذا مجرد اتفاق وما هو إلا جزء من بضعة أجـزاء XML الخاص بعدم تحسس حالة الأحرف، وذلك لوجود تراث لها في مقياس ISO الخاص بعدم تحسس حالة الأحرف.

وتماماً مثل كل السمات المستخدمة في DTD للمستندات الصالحة، يجب إعلان ســمة :XML LONG لتلك العناصر التي تطبق عليها السمة بطريقة مباشرة "تطبق هذه الســمة علــى نتــائج العناصر التي لها سمات XML:LONG محددة"، لكن تلك النتائج لا تحتاج لإعلان منفصل.

قد لا ترغب في السماح بوجود قيم اعتباطية للسمة XML: LANG محددة، لكن تلك النتـائج لا تحتاج لإعلان منفصل.

قد لا ترغب في السماح بوجود قيم اعتباطية للسمة XML: LONG وبما أن القيم المسموح بها تعد أسماء XML صـالحة، فغالباً ما تكون السمة من نوع NMTOKEN، وهذا النوع يقصر قيمـــة السمة على اسم XML صالح، مثال:

```
<!ELEMENT P (#PCDATA)>
<!ATTLIST P xml:lang NMTOKEN #IMPLIED "en">
```

وبالتعاقب لو كان هناك بضعة لغـات ولـهجات مسـموح بـها، يمكنـك اسـتخدام النـوع ENUMERATED. مثال: يتضح مما يلي من DTD أن العنصر P قد يكون إنجليزياً أو لاتينياً:

```
<!ELEMENT P (#PCDATA)>
<!ATTLIST P xml:lang (en | la) "en">
```

كما يمكنك استخدام سمة من نوع CDATA، ولكن ليست هناك أسباب كافية تستدعي ذلك، كما أن استخدام النوع ENUMERATED أو NMTOKEN يساعدك على اكتشاف بعـض الأخطـاء المحتمل الوقوع فيها.

سمة DTD مستند خاصة بإحصائية عن بلعبة البيسبول

أخرج الفصل الخامس في شكل جيد مستند XML للاتحاد الرئيسي للموسم لعـام ١٩٩٨ والتـي استخدمت سمات لتخزيـن السـنة (Year) والموسـم (Season) والاسـم (Name) الخـاص بالاتحادات والتقسيمات والفرق والمدن (City) التي أقيمت على أرضها المباريات. هذا إلى جانب إحصائيات مفصلة عن اللاعبين أنفسهم. تمثل تعليمات البرمجة ٤-١٠ الموضحة أسفل الصفحة، إصداراً أقل طولاً من تعليمات البرمجة ١-٥، فهي تمثل مستند XML كامل يحتوي على اتحادين، ستة تقسيمات وستة فرق ولاعبين وهي تساعد على إنعاش الذاكرة لمعرفة أية عناصر تنتمي لأية سمات وأين.

تعليمات البرمجة ١٠-٤: مستند XML تام

```
<?xml version="1.0" standalone="yes"?>
<SEASON YEAR="1998">
  <LEAGUE NAME="National League">
    <DIVISION NAME="East">
```

```
 <TEAM CITY="atlanta" NAME="Braves">
  <PLAYER GIVEN_NAME="marty" SURNAME="Malloy"
   POSITION="Second Base" GAMES="11" GAMES_STARTED="8"
   AT_BATS="28" RUNS="3" HITS="5" DOUBLES="1"
   TRIPLES="0" HOME_RUNS="1" RBI="1" STEALS="0"
   CAUGHT_STEALING="0" SACRIFICE_HITS="0"
   SACRIFICE_FLIES="0" ERRORS="0" WALKS="2"
   STRUCK_OUT="2" HIT_BY_PITCH="0" />
  <PLAYER GIVEN_NAME="Tom" SURNAME="Glavine"
   POSITION="Starting Pitcher" GAMES="33"
   GAMES_STARTED="33" WINS="20" LOSSES="6" SAVES="0"
   COMPLETE_GAMES="4" SHUTOUTS="3" ERA="2.47"
   INNINGS="229.1" HOME_RUNS_AGAINST="13"
   RUNS_AGAINST="67" EARNED_RUNS="63" HIT_BATTER="2"
   WILD_PITCHES="3" BALK="0" WALKED_BATTER="74"
   STRUCK_OUT_BATTER="157" />
 </TEAM>
</DIVISION>
<DIVISION NAME="central">
 <TEAM CITY="Chicago" NAME="Cubs">
 </TEAM>
</DIVISION>
<DIVISION NAME="West">
 <TEAM CITY="San Francisco" NAME="Giants">
 </TEAM>
</DIVISION>
</LEAGUE>
<LEAGUE NAME="American League">
<DIVISION NAME="East">
```

```
    <TEAM CITY="New York" NAME="Yankees">
    </TEAM>
   </DIVISION>
   <DIVISION NAME="central">
    <TEAM CITY="Minnesota" NAME="Twins">
    </TEAM>
   </DIVISION>
   <DIVISION NAME="west">
    <TEAM CITY="Oakland" NAME="Athletics">
    </TEAM>
   </DIVISION>
  </LEAGUE>
</SEASON>
```

ولكي يصير هذا المستند صالحاً، عليك باستخدام DTD والذي يجب أن يعلن الرموز والسمات المذكورة في تعليمات البرمجية ١٠-٤. وتماثل إعلانات العناصر غيرها من الإعلانات، إلا أنها أقل في العدد لأن معظم المعلومات قد تم نقلها إلى سمات:

```
<!ELEMENT SEASON (LEAGUE, LEAGUE)>
  <!ELEMENT LEAGUE (DIVISION, DIVISION, DIVISION)>
  <!ELEMENT DIVISION (TEAM+)>
  <!ELEMENT TEAM (PLAYER*)>
  <!ELEMENT PLAYER EMPTY>
```

إعلان سمات SEASON في DTD

لعنصر Season سمة واحدة وهي Year. وعلى الرغم من أن المعنى قد يحدد ما المقصود بكلمة سنة (Year)، "تعد ١٩٩٨ سنة، على عكس ٣١ مارس" فإن لا تفيد بذلك. لذا فإن أفضل الطرق تعلن أن لسمة Year أعم وأشمل أنواع السمات، أي CDATA. بالإضافة إلى ذلك، نود أن يكون لكل المواسم سنة، لذا سنجعل سمة Year مطلوبة:

```
   <!ATTLIST SEASON YEAR CDATA #REQUIRED>
```

على الرغم من أنك لن تستطيع إدخال شكل مؤلفي النص في سمات Year، فإنك على الأقل تستطيع أن تضيف تعليقاً يكشف عما هو متوقع. فعلى سبيل المثال، قد تكون فكرة جيدة لو أنك حددت أنه مطلوب أربعة أرقام للسنوات.

```
<!ATTLIST SEASON YEAR CDATA #REQUIRED> <!-- e.g. 1998 -->
  <!-- DO NOT USE TWO DIGIT YEARS like 98, 99, 00!! -->
```

إعلان سمات LEAGUE وDIVISION في DTD

فلتضع في الاعتبار بعد ذلك League وDivision، والتي لكل منهما سمة Name واحدة. مرة أخرى نقول أن النوع العام هو CDATA وستكون السمة مطلوبة. وبما أن هاتين السمتين هما سمتا Name مختلفتان لعنصرين مختلفين، سيكون مطلوباً إعلاناً <!ATTLIST> منفصلين.

```
<!ATTLIST LEAGUE   NAME CDATA #REQUIRED>
    <!ATTLIST DIVISION NAME CDATA #REQUIRED>
```

وإضافة تعليق هنا قد يساعد مؤلفي المستند على معرفة الشكل المتوقع، كأن نعد مثلاً الكلمات: اتحاد League وتقسيم (division) كجزء من الاسم أولاً نعدها.

```
<!ATTLIST LEAGUE   NAME CDATA #REQUIRED>
  <!-- e.g. "National League" -->
    <!ATTLIST DIVISION NAME CDATA #REQUIRED>
      <!-- e.g. "east" -->
```

إعلان سمات TEAM في DTD

لكلمة Team كل من Name وCity، وكل منهما يكون من نوع CDATA ويكون مطلوباً:

```
<!ATTLIST TEAM NAME CDATA #REQUIRED>
    <!ATTLIST TEAM CITY CDATA #REQUIRED>
```

وإضافة تعليق هنا قد يساعد أكثر على التوضيح، كأن نقول مثلاً أن سمة City قد يكون اسماً لولاية في بعض الأحيان.

```
<!ATTLIST TEAM NAME CDATA #REQUIRED>
  <!ATTLIST TEAM CITY CDATA #REQUIRED>
  <!-- e.g. "San Diego" as in "San Diego Padres"
      or "texas" as in "Texas Rangers" -->
```

وبالتعاقب يمكنك إعلان كلتا السمتين في إعلان <!ATTLIST> واحد.

```
<!ATTLIST TEAM NAME CDATA #REQUIRED
              CITY CDATA #REQUIRED>
```

إعلان سمات PLAYER في DTD

لعنصر Player معظم سمات Name-Given وSurname، ويعد الاثنان الأوائـــل مــن نــوع CDATA وهما مطلوبان:

```
<!ATTLIST PLAYER GIVEN_NAME CDATA #REQUIRED>
    <!ATTLIST PLAYER SURNAME    CDATA #REQUIRED>
```

وسمة Player التالية هي Position، فيما أن مراكز لعبة البيسبول قياســـية، فإنــه يمكنــك استخدام السمة ذات النوع Enumerated في هذه الحالـــة، غــير أن First Base وSecond Base وThird Base وStashing Pitcher وRelief Pitcher تحتوي جميعها على مســــافات بيضاء، لذا فهي أسماء XML غير صالحة. تبعاً لذلك لا يمكن استخدام سوى الســـمة مــن نــوع CDATA، وبما أنه لا توجد قيمة افتراضية للمركز (Position) فإن السمة تكون أيضاً مطلوبة.

```
<!ATTLIST PLAYER POSITION   CDATA #REQUIRED>
```

ثم نأتي بعد ذلـــك إلـــى الإحصائيـــات المتعـــددة مثـــل GAMES_GAMES_STARTED, AT_BATS, RUNS, HITS, WINS, LOSSES, SAVES, SHUTOUTS, وغيرها. ويجـــب بعد ذلك أن تمثل كل منها برقم ما، ولكن بما إنه لا يوجد تطبيق لطبع البيانات في XML، فإننـــا نعلمهم كسمات من نوع CDATA، وبما إنه ليس لكل اللاعبين قيمة صالحة لكل واحدة من تلـــك الإحصائيات، فإننا نعلن كلاً على حدة وتكون مفهومة ضمناً لا مطلوبة.

```
<!ATTLIST PLAYER GAMES           CDATA #IMPLIED>
  <!ATTLIST PLAYER GAMES_STARTED     CDATA #IMPLIED>

  <!- Batting Statistics ->
  <!ATTLIST PLAYER AT_BATS          CDATA #IMPLIED>
  <!ATTLIST PLAYER RUNS            CDATA #IMPLIED>
  <!ATTLIST PLAYER HITS            CDATA #IMPLIED>
  <!ATTLIST PLAYER DOUBLES          CDATA #IMPLIED>
  <!ATTLIST PLAYER TRIPLES          CDATA #IMPLIED>
  <!ATTLIST PLAYER HOME_RUNS         CDATA #IMPLIED>
  <!ATTLIST PLAYER RBI             CDATA #IMPLIED>
  <!ATTLIST PLAYER STEALS           CDATA #IMPLIED>
  <!ATTLIST PLAYER CAUGHT_STEALING   CDATA #IMPLIED>
  <!ATTLIST PLAYER SACRIFICE_HITS    CDATA #IMPLIED>
  <!ATTLIST PLAYER SACRIFICE_FLIES   CDATA #IMPLIED>
```

```
<!ATTLIST PLAYER ERRORS          CDATA #IMPLIED>
<!ATTLIST PLAYER WALKS           CDATA #IMPLIED>
<!ATTLIST PLAYER STRUCK_OUT       CDATA #IMPLIED>
<!ATTLIST PLAYER HIT_BY_PITCH     CDATA #IMPLIED>

<!- Pitching Statistics ->
<!ATTLIST PLAYER WINS            CDATA #IMPLIED>
<!ATTLIST PLAYER LOSSES          CDATA #IMPLIED>
<!ATTLIST PLAYER SAVES           CDATA #IMPLIED>
<!ATTLIST PLAYER COMPLETE_GAMES   CDATA #IMPLIED>
<!ATTLIST PLAYER SHUTOUTS        CDATA #IMPLIED>
<!ATTLIST PLAYER ERA             CDATA #IMPLIED>
<!ATTLIST PLAYER INNINGS         CDATA #IMPLIED>
<!ATTLIST PLAYER HOME_RUNS_AGAINST CDATA #IMPLIED>
<!ATTLIST PLAYER RUNS_AGAINST     CDATA #IMPLIED>
<!ATTLIST PLAYER EARNED_RUNS      CDATA #IMPLIED>
<!ATTLIST PLAYER HIT_BATTER       CDATA #IMPLIED>
<!ATTLIST PLAYER WILD_PITCHES     CDATA #IMPLIED>
<!ATTLIST PLAYER BALK            CDATA #IMPLIED>
<!ATTLIST PLAYER WALKED_BATTER    CDATA #IMPLIED>
<!ATTLIST PLAYER STRUCK_OUT_BATTER CDATA #IMPLIED>
```

يمكنك دمج كل السمات الممكنة لعنصر Player في إعلان <!ATTLIST> واحد ضخم إذا رغبت في ذلك.

```
<!ATTLIST PLAYER
  GIVEN_NAME      CDATA #REQUIRED
  SURNAME         CDATA #REQUIRED
  POSITION        CDATA #REQUIRED
  GAMES           CDATA #IMPLIED
  GAMES_STARTED   CDATA #IMPLIED
  AT_BATS         CDATA #IMPLIED
  RUNS            CDATA #IMPLIED
  HITS            CDATA #IMPLIED
  DOUBLES         CDATA #IMPLIED
```

```
TRIPLES         CDATA #IMPLIED
HOME_RUNS         CDATA #IMPLIED
RBI             CDATA #IMPLIED
STEALS          CDATA #IMPLIED
CAUGHT_STEALING   CDATA #IMPLIED
SACRIFICE_HITS    CDATA #IMPLIED
SACRIFICE_FLIES   CDATA #IMPLIED
ERRORS           CDATA #IMPLIED
WALKS           CDATA #IMPLIED
STRUCK_OUT        CDATA #IMPLIED
HIT_BY_PITCH      CDATA #IMPLIED

WINS            CDATA #IMPLIED
LOSSES           CDATA #IMPLIED
SAVES           CDATA #IMPLIED
COMPLETE_GAMES    CDATA #IMPLIED
SHUTOUTS         CDATA #IMPLIED
ERA             CDATA #IMPLIED
INNINGS          CDATA #IMPLIED
HOME_RUNS_AGAINST CDATA #IMPLIED
RUNS_AGAINST      CDATA #IMPLIED
EARNED_RUNS       CDATA #IMPLIED
HIT_BATTER       CDATA #IMPLIED
WILD_PITCHES      CDATA #IMPLIED
BALK            CDATA #IMPLIED
WALKED_BATTER     CDATA #IMPLIED
STRUCK_OUT_BATTER CDATA #IMPLIED>
```

لكن إحدى عيوب هذه الطريقة أنها تجعل من المستحيل إضافة التعليقات ولو كانت بسيطة بجانب السمات الفردية.

DTD كامل لنموذج إحصائيات خاصة بلعبة البيسبول

توضح تعليمات البرمجة ١٠-٥ الـ DTD الكامل الخاص بلعبة البيسبول استنادا ألي السمة.

تعليمات البرمجة ١٠-٥: DTD كامل للإحصائيات الخاصة بلعبة البيسبول والتي تستند إلى السمات في معظم معلوماتها

```
<!ELEMENT SEASON (LEAGUE, LEAGUE)>
<!ELEMENT LEAGUE (DIVISION, DIVISION, DIVISION)>
<!ELEMENT DIVISION (TEAM+)>
<!ELEMENT TEAM (PLAYER*)>
<!ELEMENT PLAYER EMPTY>

<!ATTLIST SEASON   YEAR CDATA #REQUIRED>
<!ATTLIST LEAGUE   NAME CDATA #REQUIRED>
<!ATTLIST DIVISION NAME CDATA #REQUIRED>
<!ATTLIST TEAM NAME CDATA #REQUIRED
CITY CDATA #REQUIRED>

<!ATTLIST PLAYER GIVEN_NAME      CDATA #REQUIRED>
<!ATTLIST PLAYER SURNAME         CDATA #REQUIRED>
<!ATTLIST PLAYER POSITION        CDATA #REQUIRED>
<!ATTLIST PLAYER GAMES           CDATA #REQUIRED>
<!ATTLIST PLAYER GAMES_STARTED   CDATA #REQUIRED>

<!- Batting Statistics ->
<!ATTLIST PLAYER AT_BATS         CDATA #IMPLIED>
<!ATTLIST PLAYER RUNS            CDATA #IMPLIED>
<!ATTLIST PLAYER HITS            CDATA #IMPLIED>
<!ATTLIST PLAYER DOUBLES         CDATA #IMPLIED>
<!ATTLIST PLAYER TRIPLES         CDATA #IMPLIED>
<!ATTLIST PLAYER HOME_RUNS       CDATA #IMPLIED>
<!ATTLIST PLAYER RBI             CDATA #IMPLIED>
<!ATTLIST PLAYER STEALS          CDATA #IMPLIED>
<!ATTLIST PLAYER CAUGHT_STEALING  CDATA #IMPLIED>
<!ATTLIST PLAYER SACRIFICE_HITS   CDATA #IMPLIED>
<!ATTLIST PLAYER SACRIFICE_FLIES  CDATA #IMPLIED>
```

```
<!ATTLIST PLAYER ERRORS          CDATA #IMPLIED>
<!ATTLIST PLAYER WALKS           CDATA #IMPLIED>
<!ATTLIST PLAYER STRUCK_OUT      CDATA #IMPLIED>
<!ATTLIST PLAYER HIT_BY_PITCH    CDATA #IMPLIED>

<!- Pitching Statistics ->
<!ATTLIST PLAYER WINS            CDATA #IMPLIED>
<!ATTLIST PLAYER LOSSES          CDATA #IMPLIED>
<!ATTLIST PLAYER SAVES           CDATA #IMPLIED>
<!ATTLIST PLAYER COMPLETE_GAMES  CDATA #IMPLIED>
<!ATTLIST PLAYER SHUTOUTS        CDATA #IMPLIED>
<!ATTLIST PLAYER ERA             CDATA #IMPLIED>
<!ATTLIST PLAYER INNINGS         CDATA #IMPLIED>

<!ATTLIST PLAYER HOME_RUNS_AGAINST CDATA #IMPLIED>
Continued
Listing 10-5ت(continued)
<!ATTLIST PLAYER RUNS_AGAINST    CDATA #IMPLIED>
<!ATTLIST PLAYER EARNED_RUNS     CDATA #IMPLIED>
<!ATTLIST PLAYER HIT_BATTER      CDATA #IMPLIED>
<!ATTLIST PLAYER WILD_PITCHES    CDATA #IMPLIED>
<!ATTLIST PLAYER BALK            CDATA #IMPLIED>
<!ATTLIST PLAYER WALKED_BATTER   CDATA #IMPLIED>
<!ATTLIST PLAYER STRUCK_OUT_BATTER CDATA #IMPLIED>
```

لربط ما ذكر بأعلى بتعليمات البرمجة ١٠-٤، استخدام البرولوج التالي الـــذي يفـــترض أن النموذج ١٠-٥ مخزن في ملف يحمل اسم baseball attributes. dtb.

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE SEASON SYSTEM "baseballattributes.dtd" >
```

خلاصة

لقد تعرفت في هذا الفصل على كيفية إعلان سمات العناصر في DTD وبخاصة تعرفـت علـى الأفكار التالية:

- تعلن السمات في DTD في علامة <!ATTLIST>.
- يمكن لعلامة <!ATTLIST> واحدة أن تعلن للعنصر الواحد عددا غـير محـدود مـن السمات.
- عادة ما يكون للسمات قيم افتراضية لكن قد يتبدل هذا الوضع باستخدام الكلمات الأساسية REQUIRED أو IMPLIED أو FIXED.
- هناك عشرة أنـواع مختلفـة للسـمات يمكـن إعلانـها فـي DTD وهـي CDATA وENUMERARED وNMTOKEN وNMTOKEN وID وIDREFS وENTITY وENTITIES وNOTATION.
- سمة XML: SPACE المعرفة من قبل تحدد مدى أهمية المسافة البيضاء في عنصر ما.
- سمة XML: LANG المعرفة من قبل تحدد لغة محتوى العنصر.

في الفصل اللاحق ستتعرف على كيفية استخدام الملاحظات والتعليمات المعالجة والكيانـات الخارجية غير المحللة في تضمين بيانات ليست من نوع XML داخل مستندات XML.

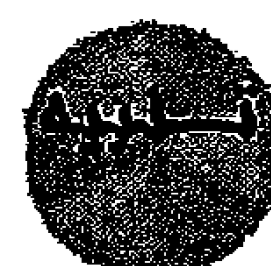


إضافة بيانات ليست من نوع XML

في هذا الفصل

ليست كل البيانات في العالم من نوع XML بل يمكن القول بأن معظمها لا ينتمي إلى هذا النــوع، لكن يخزن الكثير منها في نص عادي أو HTML أو مايكروسوفت وورد، وهذه هي أكثر ثــلاث صيغ "عدا XML" شيوعاً وعلى الرغم من أن معظم هذه البيانات يمكن اعتبارها على الأقل مـــن الناحية النظرية من نوع XML من حيث المصادر والزيادة المسموح بهما فإنه لا يجب أن تكــون جميع البيانات في كل أنحاء العالم من نوع XML. إضافة إلى ذلك فإن تحويل الصور إلى رمــوز في XML لن يعطي نتيجة.

يوفر XML ثلاث شاشات، عادة ما تستخدم عند تناول بيانات من غــير نــوع XML وهــي: الملاحظات، الكيانات الخارجية غير الموزعة والتعليمات المعالجة. تعطي الملاحظـــات وصفــاً بصيغة البيانات من غير نوع XML، بينما تحدث الكيانات الخارجيــة غــير الموزعــة اتصـــالاً بالبيانات من غير XML، وأخيراً تمد التعليمات المعالجة بالمعلومات الكافية عن كيفيـــة عــرض البيانات.



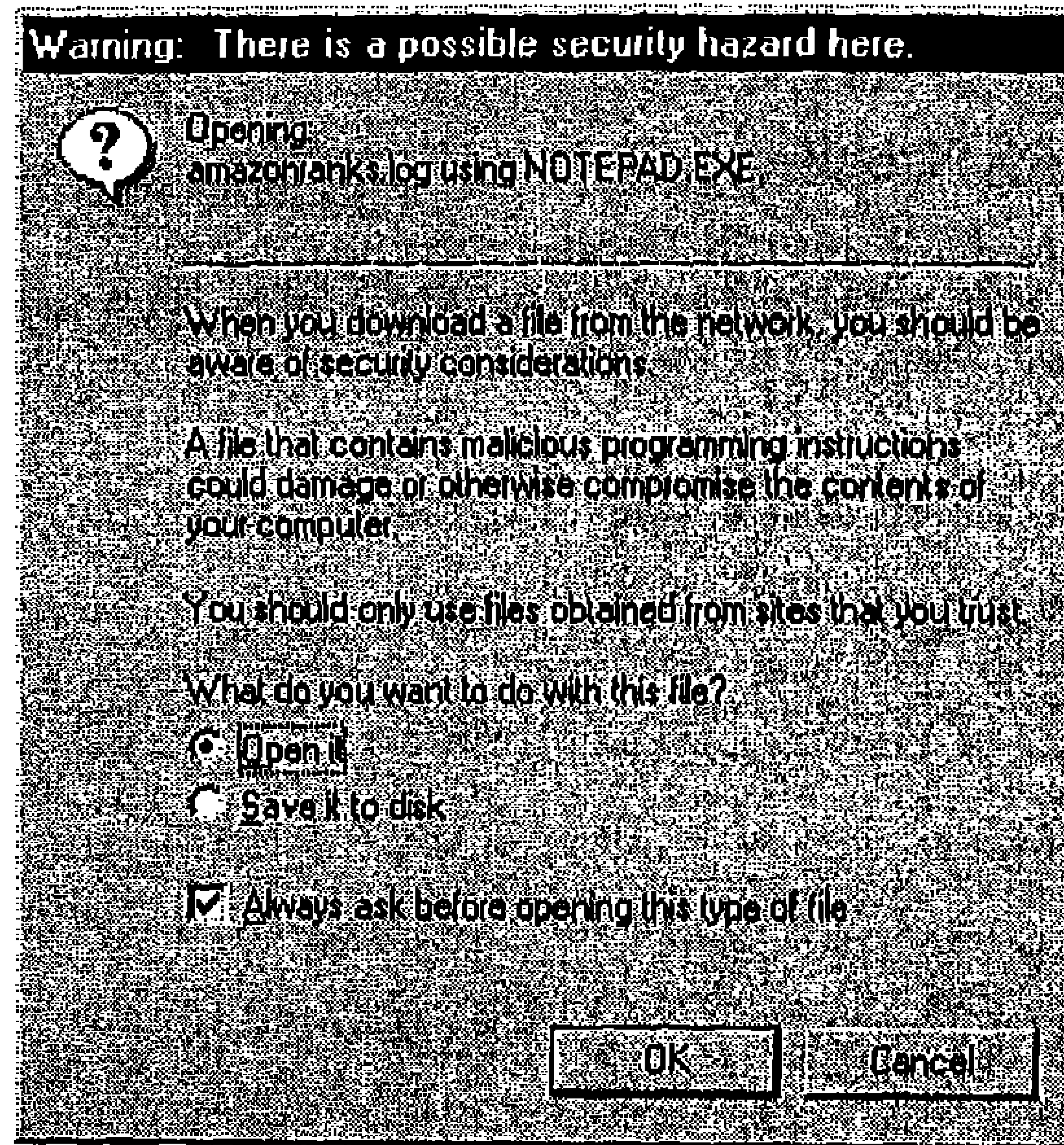
تعد المادة المتناولة في هذا الفصل مثاراً للجدل، فعلى الرغم من أن كل ما تناولته بالشرح هو جزء من XML 1.0 فإن هناك من لا يوافقون على ذلك، حيــث أنــه يمكنك كتابة مستندات XML دون استخدام الملاحظات أو الكيانات الخارجية غــير الموزعة، لكن فقط باستخدام بضعة تعليمات معالجة. يمكنك أن تخطي هذا الفصل جانباً ثم تعود لقراءته مرة أخرى بعد ذلك إذا كانت هناك حاجة ملحة لذلك.

الملاحظات

من أولى المشكلات التي تواجهها عند التعامل مع بيانات من غير نوع XML في مســـتند XML، هي تحديد صيغة هذه المعلومات وتوضيح لتطبيق XML كيفية قــــراءة وعرض بيانـــات XML، فستجد مثلاً مشكلة عند محاولة عرض ملف صوت MP3 على الشاشة.

لكن يمكنك حل تلك المشكلة بتطبيق واحد وذلك باستخدام مجموعة ثابتة العلامـــات لبعــض أنواع الكيانات الخارجية. مثال: لو تم إدخال جميع الصور ضمــن عنــاصر IMAGE، وجميــع الأصوات ضمن عناصر AUDIO، فلن يمكن من الصعب تطوير عارض يعرف كيف يتعامل مع هذين العنصرين، وهذه هي الطريقة التي تتبعها HTML. غير أن هذه الطريقة لا تمنــع مؤلفــي المستند من إنشاء علامات جديدة أكثر تحديداً بحيث يكون لها قدرة أكبر على وصف المحتــوى، ومثال على ذلك عنصر PERSON الذي، سمة PHOTO التي تشير إلى صــورة JPEG لذلــك الشخص.

إضافة إلى ذلك، لا يوجد تطبيق يفهم كل صيغ الملفات الممكنـــة، وبينمـــا تســـتطيع معظـــم عارضتي ويب تمييز وقراءة GIF وJPEG وPNG وربما أنواعاً أخرى من ملفات الصـــور، إلا أنها لا تظهر نفس القدرة مع ملفات EPS وTIFF وFITS ومئات غيرها مـــن صيـــغ الصـــور. والحوار الظاهر أمامك في شكل ١١-١ يبدو مألوفا.



الشكل ١١-١ مـــا يظـــهر عندمـــا لا يســـتطيع Netscape Navigator التعـــرف على نوع الملف.

ولعله من الأفضل أن يكون لديك مستندات لتمد التطبيق بالمعلومات اللازمة عن صيغة الكيان الخارجي فلا تضطر إلى الاعتماد على التطبيق الذي يتعرف على نوع الملف وذلـــك باســـتخدام رقماً سحرياً أو باستخدام ملحق اسم الملف الذي لا يمكنه الاعتماد عليه. كما أنك قد تريد أن تشير في التطبيق إلى ما يستخدمه البرنامج لعرض الصورة وذلك في حالة عدم عرضها تلقائياً.

وتقدم الملاحظات حلاً جزئياً "لكن لا يكون دائما مدعماً" لتلك المشكلة، حيث تصـــف صيغـــة البيانات من غير نوع XML ويحدد إعلان NOTATION في DTD نوعاً معيناً للبيانات. كما تعلن DTD في نفس الوقت الملاحظات تماماً مثل العناصر والسمات والكيانات ويحتوي كـــل إعـــلان ملاحظة على اسم ومعرف خارجي تبعاً لبناء الجملة التالي:

```
<!NOTATION name SYSTEM "externalID">
```

ويعد الاسم "name" معرفاً لتلك الصيغة المستخدمة في المستند كما يحتوي الرقم المعرف الخارجي "externalID" على سلسلة بشرية واضحة تقوم بتعريف الملاحظة، فعلى سبيل المثال يمكنك استخدام أنواع MIME كتلك المستخدمة في هذه الملاحظة لصور GIF.

```
<!NOTATION GIF SYSTEM "image/gif">
```

كما يمكنك أيضا استخدام المعرف PUBLIC بدلاً من SYSTEM، لكن بشرط أن توفر ID عام وURL. مثال:

```
<!NOTATION GIF PUBLIC
  "-//IETF//NONSGML Media Type image/gif//EN"
  "http://www.isi.edu/in-notes/iana/assignments/media-
types/image/gif">
```



هناك جدل حول ما يجعل من المعرف الخارجي معرفاً جيداً، فالبعض يرى أن الفضل في ذلك يكون لأنواع MIME مثل صورة GIF أو نص HTML، بينما يرى آخرون أنها أنواع URLs أو مواقع أخرى للمستندات القياسية مثل http://www.w3.org/TR/REC-html40/. بينما يخص الرأي الثالث بالفضل اسم قياس عالمي رسمي مثل ISO 8601 الذي يمثل التواريخ والمواعيد. وفي بعض الأحيان يكون رقم النشرة المصورة الخاصة بمكتبة الكونجرس أو ISBN لورقة المستند التي يكون من الاختيارات الأخرى.

وسيأتي اختيارك معتمداً على الدورة الحياتية المتوقعة للمستند، فإذا استخدمت مثلاً صيغة غير عادية فإنك لن تكون بحاجة إلى الاعتماد على URL الذي يتغير من شهر إلى شهر وإذا كنت ترغب في أن يظل المستند باقياً جديداً بالاهتمام على مدار مائة عام، فعليك اختيار المعرفات التي ستظل معانيها قائمة حتى بعد مائة عام.

وللحديث عن البيانات التي لا تتناسب مع مستند XML يمكنك استخدام الملاحظات، مثال على ذلك عنصر DATE:

```
<DATE>05-07-06</DATE>
```

هذا المثال يجعلنا نتساءل، أي يوم تحديداً يمثله التاريخ: ٠٥-٠٧-٠٦ هل هو السابع من مايو عام ١٩٠٦، أم هو الخامس ممن يوليو عام ١٩٠٦؟ للإجابة عن هذا السؤال يجب تحديد أولاً ما إذا كنت تقرأ هذا التاريخ في الولايات المتحدة أو في أوروبا. فهناك احتمالات أخرى لقراءة هذا التاريخ، قد يكون السابع من مايو عام ٢٠٠٦، أو الخامس من يوليو عام ٢٠٠٦، أو حتى السابع

من مايو عام ٦ أي أثناء فترة حكم الإمبراطور الروماني أغسطس للغرب. ومن المحتمل أيضا ألا يكون هذا التاريخ للتقويم المتداول بل يكون مكتوبا بالتقويم اليهودي، الإسلامي أو الصيني. لذا بدون معلومات إضافية لن تستطيع معرفة المعنى الحقيقي لذلك التاريخ.

ولتجنب الوقوع في مشكلة من هذا النوع، يحدد قياس ISO 8601 وسيلة دقيقة تمثل التواريخ، وتبعاً لها يكتب التاريخ ٥ يوليو ٢٠٠٦ كالأتي: ٢٠٠٦٠٧٠٥، أو يكتب في XML كالاتي:

```
<DATE>20060705</DATE>
```

إلا أن هذه الصيغة لا تعتبر واضحة للجميع وهي لذلك لا تعتبر عن حضارة معينة "على الرغم من تحيزه للتقويم الغربي التقليدي".

تعلن الملاحظات في DTD وتستخدم في سمات الملاحظات لوصف تنسيق البيانات من غير نوع XML والتي تضمن في مستند XML واستكمالاً للحديث عن المثال الخاص بالتاريخ بعرض تعليمات البرمجة ١١-١ والتي تعرف اثنين من الملاحظات الممكنة استخدامها للتواريخ في قياس ISO 8601 وفي تنسيقات الولايات المتحدة الاصطلاحية ثم تضاف بعد ذلك سمة FORMAT مطلوبة من نوع NOTATION إلى كل عنصر DATE لوصف البناء الخاص بعنصر معين.

تعليمات البرمجة ١١-١: عناصر DATE في قياس ISO 8601 وفي تنسيقات الولايات المتحدة الاصطلاحية

```
<?xml version="1.0": standalone="yes"?>
<!DOCTYPE SCHEDULE [
  <!NOTATION ISODATE SYSTEM
    "http://www.iso.ch/cate/d15903.html">
  <!NOTATION USDATE SYSTEM
"http://es.rice.edu/ES/humsoc/Galileo/Things/gregorian_calendar.html"
>

  <!ELEMENT SCHEDULE (APPOINTMENT*)>
  <!ELEMENT APPOINTMENT (NOTE, DATE, TIME?)>

  <!ELEMENT NOTE (#PCDATA)>
  <!ELEMENT DATE (#PCDATA)>
```

```
  <!ELEMENT TIME (#PCDATA)>
<!ATTLIST DATE FORMAT NOTATION (ISODATE | USDATE)
#IMPLIED>

]>
<SCHEDULE>
  <APPOINTMENT>
    <NOTE>Deliver presents</NOTE>
    <DATE FORMAT="USDATE">12-25-1999</DATE>
  </APPOINTMENT>
  <APPOINTMENT>
    <NOTE>Party like it's 1999</NOTE>
    <DATE FORMAT="ISODATE">19991231</DATE>
  </APPOINTMENT>
</SCHEDULE>
```

ولا تفرض الملاحظات على المؤلفين استخدام الصيغة التي تصفها لكن لتحقيق ذلك عليك باستخدام مخطط للغة إلى جانب XML، غير أنه في حالة الاستخدامات البسيطة يكون كافياً أن يقوم المؤلفون بوصف البيانات وصفاً صحيحاً.

الكيانات الخارجية غير الموزعة

لا تعتبر XML صيغة مثاليه لكل البيانات، وخاصة البيانات غير النصية. فعلى سبيل المثال يمكنك تخزين كل بكسل لصورة نقطية كعنصر XML كالتالي.

```
<PIXEL X="232" Y="128" COLOR="FF5E32" />
```

وهذه ليست بالفكرة السيئة، غير أن ذلك قد يصل بملفات الصور إلى أحجام ضخمة. وبما أنك لا تستطيع تحويل كل البيانات إلى رموز في XML، فيجب أن تشير مستندات XML إلى بيانات ليست من نوع XML ولن تكون كذلك في يوم ما.

وقد تتضمن صفحة ويب النموذجية صور GIF وJPEG، وتطبيقات Java الصغيرة، ووحدات تحكم ActiveX وأنواع عديدة من الأصوات إلى غير ذلك. وفي XML يطلق على البيانات المقتبسة من غير نوع XML كيان غير موزع وذلك لأن معالج XML لن يستطيع فهمه، بل على الأكثر يقوم بإخبار التطبيق بوجود الكيان ويمد التطبيق باسمه ومن الممكن "وليس من الضروري" بمحتواها.

وتضمن صفحات HTML الكيانات من غير هذا النوع عن طريق مجموعــــة متنوعــة مــن العلامات المخصصة، وتكون الصور ضمن علامة <IMG> التي لها سمة SRC تقــوم بتوفــير URL الخاصة بملف الصورة. أما التطبيقات الصغير ة فهي تضمن عبر علامــــة <APPLET> التي تشير سماتها CLASS وCODEBASE إلى الملف والدليل حيث يكمن التطبيق الصغير. وعن علامة <OBJECT> يمكن القول بأنها تستخدم سمة codebase لتشير إلى URI حيــث توجــد بيانات الكائن. وفي جميع الأحوال تكون هناك علامة معينة معرفة من قبل تمثــل نوعــاً معينــاً للمحتوى وتحتوي سمة معرفة من قبل على URI لذلك المحتوى.

يمكن أن تعمل تطبيقات XML بهذا الشكل لكنه ليس من الضروري. في واقع الأمر،أغلب تلك التطبيقات لا تعمل بهذا الشكل إلا إذا كانت تحاول المحافظة على الإبقاء على بعض المتوافقــات الإرتجاعية مع HTML لذا بدلاً من ذلك تستخدم تطبيقات XML كيان خارجي غير موزع للإشارة إلى المحتوى، ومثل هذه الكيانات تحقق الربط بالموقع الأصلي للبيانات من غير نوع XML ثم يتم بعد ذلك استخدام سمة من نوع ENTITY لربط ذلك الكيان بعنصر معين في المستند.

إعلان الكيانات غير الموزعة

بناء على ما جاء في الفصل التاسع، يكون إعلان الكيان الخارجي أقرب إلى هذا الشكل:

```
<!ENTITY SIG SYSTEM "http://metalab.unc.edu/xml/signature.xml">
```

غير أن هذا الشكل لا يكون مقبولاً إلا إذا كان الكيان الخارجي لأسماء URL مستند XML جيد الشكل. أما إذا كان الكيان الخارجي ليس من نوع XML، فإنه يتعين عليك تحديد نــوع الكينونــة مستخدماً الكلمة الأساسية NDATA فعلى سبيل المثال لربط ملف GIF logo.gif بالاسم LOGO، تستطع إعلان ENTITY التالي في DTD.

```
<!ENTITY LOGO SYSTEM "logo.gif" NDATA GIF>
```

يجب أن تكون أخر كلمة في الإعلان، وهي GIF في هذا المثال، اسما لملاحظة أعلنت فــي DTD، وتقوم الملاحظات بربط الاسم، GIF مثلاً، بمعرف خارجي للصيغة مثل MIME أو قياس ISO أو URL لمواصفة خاصة بالصيغة. مثال يكون شكل ملاحظة GIF كالآتي.

```
<!NOTATION GIF SYSTEM "image/gif">
```

وكالمعتاد، يمكنك استخدام URL مطلقة أو نسبية كأوامر اصطلاحية للكيان الخارجي، مثال:

```
<!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
  NDATA GIF>
<!ENTITY LOGO SYSTEM "/xml/logo.gif" NDATA GIF>
<!ENTITY LOGO SYSTEM "../logo.gif" NDATA GIF>
```

تضمين الكيانات غير الموزعة

لا يمكنك تضمين كيانات غير موزعة بنفس سهولة تضمين كيانات موزعة في موقع تلقائياً في المستند باستخدام كيان مرجعي عام لذلك نجد أن تعليمات البرمجة ١١-٢ مثلاً هي مستند XML غير صالح لأن LOGO كيان غير موزع. ربما أصبح صالحاً لو أن LOGO كانت كياناً موزعاً.

تعليمات البرمجة ١١-٢: مستند XML غير صالح يحاول تضمين كيان غير موزع كينونة مرجعي عام

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT ANY>
  <!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
    NDATA GIF>
  <!NOTATION GIF SYSTEM "image/gif">
]>
<DOCUMENT>
  &LOGO;
</DOCUMENT>
```

لتضمين كينونات غير موزعة دون استخدام كينونات مرجعية عامة مثل &LOGO، يمكنك إعلان عنصر ما يكون بمثابة حرفاً نائباً عن الكيان غير الموزع "مثل IMAGE على سبيل المثال" ثم أعلن بعد ذلك سمة من نوع ENTITY لعنصر IMAGE، مثل SOURCE على سبيل المثال، التي تعطي فقط اسم الكيان غير الموزعة. وتوضح تعليمات البرمجة ١١-٣ ذلك.

تعليمات البرمجة ١١-٣: مستند XML صالح يقوم بتضمين كيان غير موزع بطريقة سليمة

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT ANY>
  <!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
```

```
   NDATA GIF>
  <!NOTATION GIF SYSTEM "image/gif">
  <!ELEMENT IMAGE EMPTY>
  <!ATTLIST IMAGE SOURCE ENTITY #REQUIRED>

]>
<DOCUMENT>
  <IMAGE SOURCE="LOGO" />
</DOCUMENT>
```

الكيان الآن يقوم التطبيق بقراءة مستند XML للتعرف على الكيان غير الموزع وعرضه، لكن قد لا تستطيع بعض التطبيقات عرضه "تماماً مثلما يختار عارض ويب عدم تحميل الصور عندما يقوم المستخدم بتعطيل تحميل الصور".

وتصور الأمثلة التالية العناصر الخالية كحاويات غير الموزعة، غير أن ذلك لا يكون دائماً ضرورياً. تخيل مثلاً نظام ID المشترك XML والذي يستخدمه رجال الأمن لمراقبة كل من يدخل إلى المبنى، ستجد أن لعنصر PERSON سمات NAME، PHONE، OFFICE، وأطفال EMPLOYEE_ID و PHOTO ENTITY وتوضح تعليمات البرمجة ١١-٤ ذلك:

تعليمات البرمجة ١١-٤: عنصر PERSON غير خالي به سمة PHOTO ENTITY

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE PERSON [
  <!ELEMENT PERSON (NAME, EMPLOYEE_ID, PHONE, OFFICE)>
  <!ELEMENT NAME        (#PCDATA)>
  <!ELEMENT EMPLOYEE_ID  (#PCDATA)>
  <!ELEMENT PHONE        (#PCDATA)>
  <!ELEMENT OFFICE       (#PCDATA)>
  <!NOTATION JPEG SYSTEM "image/jpg">
  <!ENTITY ROGER SYSTEM "rogers.jpg" NDATA JPEG>
 <!ATTLIST PERSON PHOTO ENTITY #REQUIRED>

 ]>
 <PERSON PHOTO="ROGER">
```

```
<NAME>Jim Rogers</NAME>
<EMPLOYEE_ID>4534</EMPLOYEE_ID>
<PHONE>X396</PHONE>
<OFFICE>RH 415A</OFFICE>
</PERSON>
```

قد يبدو هذا المثال متكلفاً "مصطنعا" بعض الشيء، لذلك عند التطبيق ننصح بعمـــل عنصـــر PHOTO خالي به سمة SOURCE كنتيجة لعنصر PERSON لا كسمة له، وبتقسيم DTD إلـــى مجموعات جزئية خارجية وداخلية وتوضح المجموعة الجزئية الخارجية، كما هو موضـــح فـــي تعليمات البرمجة ١١-٥، العناصر والملاحظات والسمات وهي الأجزاء التـــي تشـــترك فيـــها. المستندات المختلفة، غير أن الكيان يتغير من مستند إلى أخر، لذا يكون من الأفضل وضعه ضمن مجموعة DTD الجزئية الداخلية للمستند كما هو موضح بتعليمات البرمجة ١١-٦.

تعليمات البرمجة ١١-٥: مجموعة DTD الجزئية الخارجية person.dtd

```
<!ELEMENT PERSON (NAME, EMPLOYEE_ID, PHONE, OFFICE,
PHOTO)>
  <!ELEMENT NAME        (#PCDATA)>
  <!ELEMENT EMPLOYEE_ID  (#PCDATA)>
  <!ELEMENT PHONE       (#PCDATA)>
  <!ELEMENT OFFICE      (#PCDATA)>
  <!ELEMENT PHOTO        EMPTY>
  <!NOTATION JPEG SYSTEM "image/jpeg">
  <!ATTLIST PHOTO SOURCE ENTITY #REQUIRED>
```

تعليمات البرمجة ١١-٦: مستند له عنصر PERSON غير خالي ومجموعة DTD الجزئية الداخلية

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE PERSON [
<!ENTITY % PERSON_DTD SYSTEM "person.dtd">
  %PERSON_DTD;
  <!ENTITY ROGER SYSTEM "rogers.jpg" NDATA JPEG>
```

```
]>
<PERSON>
  <NAME>Jim Rogers</NAME>
  <EMPLOYEE_ID>4534</EMPLOYEE_ID>
  <PHONE>X396</PHONE>
  <OFFICE>RH 415A</OFFICE>
  <PHOTO SOURCE="ROGER"/>
</PERSON>
```

تضمين كيانات متعددة غير محللة

هناك بعض الحالات النادرة التي تستدعي الإشارة إلى أكثر من كيان غير محلل، وربما إلى عدد غير محدد، للسمة الواحدة ويمكنك تحقيق ذلك بإعلان سمة كيان الحرف النائب سمة من نوع ENTITIES، والتي تتكون قيمتها من أسماء متعددة لكيانات غير محللة يفصل بينها مسافة ويشير كل اسم كيان إلى مصدر بيانات خارجي من غير نوع XML والتي يجب إعلانها في DTD ويمكنك مثلاً استخدام ذلك لكتابة عرض الشريحة التي تعرض صوراً مختلفة وسيتطلب DTD تلك الإعلانات.

```
<!ELEMENT SLIDESHOW EMPTY>
<!ATTLIST SLIDESHOW SOURCES ENTITIES #REQUIRED>
<!NOTATION JPEG SYSTEM   "image/jpeg">
<!ENTITY CHARM SYSTEM    "charm.jpg"   NDATA JPEG>
<!ENTITY MARJORIE SYSTEM "marjorie.jpg" NDATA JPEG>
<!ENTITY POSSUM SYSTEM   "possum.jpg"   NDATA JPEG>
<!ENTITY BLUE SYSTEM     "blue.jpg"     NDATA JPEG>
```

وعندما تصل إلى اللحظة التي تود فيها أن يظهر عرض الشريحة في المستند، قم بإدراج العلامة التالية.

```
<SLIDESHOW SOURCES="CHARM  MARJORIE POSSUM BLUE">
```

مرة أخرى أود التأكيد على أن هذا ليس بالوصفة السحرية التي يفهمها كل عارض XML تلقائياً، بل هو ببساطة إحدى تقنيات العارض، وقد تضمن أولاً تضمن التطبيقات الأخرى بيانات من غير نوع XML في المستندات.

التعليمات المعالجة

في كثير من الأحيان تهمل التعليقات وتظلم من أجل دعم ملاحق HTML المسجلة مثـــل جـــانب الملقم والذي يتضمن لغات استعراض أشكال خطوط محددة، قوالب قاعدة البيانات وإلى غير ذلك من العديد من العناصر المأخوذة من خارج نطاق قياس HTML والميزة من استخدام التعليقـــات لتلك الأعراض هو إعطاء الأنظمة الأخرى فرصة تجاهل البيانات العرضية غير المفهومة، بينمـا يكمن العيب في إن المستند المجرد من التعليقات قد يتغير تغيراً جذرياً، وقد يتـــم معالجـــة تلـــك التعليقات التي من المفترض أن تكون وثائق، بدون قصد، كإدخال لتلك الملاحق المسجلة ولتجنب هذا الإهمال للتعليقات، يقدم XML التعليمات المعالجة وهي تقنيـــة واضحـــة تســـتخدم لتضميـــن المعلومات داخل ملف الهدف منه إجراء تطبيقات مسجلة لا تحليل XML، كما تقـــدم معلومـــات إضافية عن كيفية عرض الكيانات الخارجية غير المحللة.

والتعليمات المعالجة هي سلسلة لنص يكون بين علامتي ?> و<? والشيء الوحيد المطلـــوب بالنسبة لبناء الجملة في نص داخل التعليمات المعالجة هو ضرورة بدء الجملة باســـم XML ثـــم إضافة مسافة ثم إضافة البيانات. وقد يكون اسم XML هو الاسم الأصلي "الحقيقي" للتطبيق "مثـــل latex" أو اسم ملاحظة في DTD تشير إلى التطبيق "مثل LATEX" حيـــث تعلـــن LATEX فـــي DTD بهذا الشكل.

```
<!NOTATION LATEX SYSTEM "/usr/local/bin/latex">
```

وقد يكون أسماء يتم التعرف عليه بواسطة تطبيق يحمل اسماً مختلفاً، وتكون التفاصيل خاصة تحديداً بالتطبيق المراد للتعليمات المعالجة للتعامل معه. وفي واقـــع الأمـــر، ســـتفرض معظـــم التطبيقات التي تعتمد على التعليمات المعالجة تركيباً أكبر على محتويات التعليمية المعالجة انظـــر مثلاً إلى تلك التعليمات المعالجة المستخدمة في لغـــة Bean Markup Language الخاصـــة IBM:

```
<? bmlpi register demos.calculator.EventSourceText2Int?>
```

اسم التطبيق المراد من تلك التعليمات هو bmlpi، والبيانات المعطاة لذلك التطبيق هي سلسـلة register demos.calculator EventSourceText2Int والتي تضمن الحزمة الكاملة لاســـم مؤهل لفئة Java، فيعطي ذلك إشارة البدء للتطبيق الذي يحمل اسم bmlpi لاستخدام فئـــة جافـــا demos.calculator.EventSourceText2Int لتحويل أحداث الإجراءات إلى إعداد فإذا حدث أن واجه bmlpi التعليمات المعالجة أثناء قراءة المستند، فإنه سيقوم علـــى الفـــور بتحمـــل فئـــة demos.calculator.EventSourceText2Int وتحويل الأحداث إلى إعداد.

قد يبدو ذلك مفصلاً ومحدداً بعض الشيء، لكنه بالفعل كذلك، فلا تعتبر التعليمات المعالجـــة جزءاً من البناء العام للمستند، بل هي تهدف إلى توفير معلومات مفصلة إضافية لبعض التطبيقات

المعينة وليس لأي تطبيق يستطيع قراءة المستند ولو حدث أن واجه تطبيق أخر تلـــك التعليمـــات أثناء قراءة المستند، فإنه سيتجاهلها بكل بساطة.

ويمكن للتعليمات المعالجة أن تتخذ أي موضع في مستند XML، إلا أن تكون داخل علامـــة أو مقطع CDATA، لذا فقد تظهر في البرولوج أو في DTD في محتوى إحدى العناصر، أو حتـــى بعد علامة إغلاق المستند ولان التعليمات المعالجة لا تعد عناصر، فهي لا تؤثـــر علـــى البنـــاء المتفرع للمستند، ومن لا يستلزم الأمر منك أن تفتح أو تغلق التعليمات المعالجة أو حتى أن تقلــق بشأن تداخلها مع عناصر أخرى، فالتعليمات المعالجة ليست علامات ولا تحد من العناصر.

ولعلك تعرف بالفعل إحدى نماذج التعليمات المعالجة، وهـــي xml-stylesheet المســـتخدمة لربط أوراق النمط بالمستندات:

```
<?xml-stylesheet type="text/xsl" href="baseball.xsl"?>
```

وعلى الرغم من أن هذه الأمثلة تظهر في البرولوج الخاص بالمستند فإنها قـــد تظـــهر فـــي التعليمات المعالجة العامة في أي مكان بالمستند وليست هناك حاجـــة لإعـــلان تلـــك التعليمـــات كعناصر ناحية عن العنصر الذي يحتوى على تلك التعليمات حيث أنها ليست عناصر في ذاتها.

وعند التعليمات المعالجة التي تبدأ بسلسلة xml، فيكون لها بعض الاستخدامات المعرفة فـــي قياس XML وإلا فلك مطلق الحرية لاستخدام أي اسم وأي سلسلة نص داخل تعليمات معالجة مـــن دون سلسلة الغلق "الإنهاء" <?، فجميع الأمثلة التالية هي تعليمات معالجة صالحة.

```
<?gcc HelloWorld.c ?>
<?acrobat document="passport.pdf"?>
<?Dave remember to replace this one?>
```



ليس من الضروري أن يستفيد معالج XML من تلك التعليمات، بل هو يقوم فقـــط بتمريرها إلى التطبيق الذي يقرر بدوره كيف يتصرف بها. ومعظـــم التطبيقـــات تتجاهل التعليمات المعالجة غير المفهومة.

في بعض الأحيان تكون معرفة نوع الكينونة الخارجية غير الموزعة لا تكفي ومن ثم يتحتـــم عليك معرفة أي برنامج تدير لعرض الكيان وأي معلمات تحتاج لتوفيرها لذلـــك البرنـــامج هنـــا يمكنك استخدام تعليمات معالجة لتوفير تلك المعلومة، فبما أن التعليمات المعالجة تحتـــوي علـــى بيانات تلقائية فإنه يكون سهلاً نسبياً عليها أن تحتوى على تعليمات تحدد الأجزاء التي يجـــب إن يأخذها البرنامج الخارجي المذكور في الملاحظة.

تتجاوز التعليمات المعالجة كونها اسماً لبرنامج يمكنه عرض الملف ليصل إلى وحدات عديـــدة من الكيلو بايت للمعلومات التكوينية ويجب أن يستخدم كل من التطبيق ومؤلـــف المســـتند نفـــس

الوسائل لتحديد أية تعليمات معالجة تنتمي إلى أية كيانات خارجية غير محللة. وتوضح تعليمـــات البرمجة ١١-٧ إحدى النظم التي تستخدم تعليمات معالجة، وملاحظة PDF لمحاولة نقل إصـــدار PDF لورقة فيزياء إلى Acrobat Reader لعرضها.

تعليمات البرمجة ١١-٧: تضمين مستند PDF في XML

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE PAPER [

 <!NOTATION PDF PUBLIC
   "-//IETF//NONSGML Media Type application/pdf//EN"
   "http://www.isi.edu/in-notes/iana/assignments/media-
types/application/pdf">

  <!ELEMENT PAPER (TITLE, AUTHOR+, JOURNAL, DATE_RECEIVED,
VOLUME, ISSUE, PAGES)>
  <!ATTLIST PAPER CONTENTS ENTITY #IMPLIED>
  <!ENTITY PRLTAO000081000024005270000001 SYSTEM

"http://ojps.aip.org/journal_cgi/getpdf?KEY=PRLTAO&amp;cvips=PRLTA
O000081000024005270000001"
   NDATA PDF>

  <!ELEMENT AUTHOR (#PCDATA)>
  <!ELEMENT JOURNAL (#PCDATA)>
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT DATE_RECEIVED (#PCDATA)>
  <!ELEMENT VOLUME (#PCDATA)>
  <!ELEMENT ISSUE (#PCDATA)>
  <!ELEMENT PAGES (#PCDATA)>

]>
```

```
<?PDF acroread?>
<PAPER CONTENTS="PRLTAO000081000024005270000001">
  <TITLE>Do Naked Singularities Generically Occur in Generalized
Theories of Gravity?</TITLE>
  <AUTHOR>Kengo Maeda</AUTHOR>
  <AUTHOR>Takashi Torii</AUTHOR>
  <AUTHOR>Makoto Narita</AUTHOR>
  <JOURNAL>Physical Review Letters</JOURNAL>
  <DATE_RECEIVED>19 August 1998</DATE_RECEIVED>
  <VOLUME>81</VOLUME>
  <ISSUE>24</ISSUE>
  <PAGES>5270-5273</PAGES>
</PAPER>
```

كالمعتاد، يجب أن تضع في الاعتبار أنه ليس كل معالج سيتعامل مع هذا المثـــال بالطريقــة المرادة. بل أن أغلبهم لن يفعل، إلا أن هذه هي إحدى النظم التي يمكن أن تعبر عن كيفية دعـــم التطبيق لملفات PDF وأنواع وسائط أخرى من غير نوع XML.

المقاطع الشرطية في DTDs

عند القيام بتوضيح DTDs أو مستندات، فإنه يتعين عليك ملاحظة بعض الأجزاء غير الظـــاهرة في المستندات. وبالإضافة إلى استخدام التعليقات بطريقة مباشرة، يمكن حذف مجموعة بعينها من الإعلانات الموجودة في DTD وذلك بإخفائها في تعليمات IGNORE، فيكون بناء الجملة كالأتي.

```
<![ IGNORE
  declarations that are ignored
]]>
```

وكما نقول دائماً أن وجود مسافة لا يؤثر على بناء الجملـــة، لكـــن يجـــب وضـــع البدايـــة <![IGNORE والنهاية Closing]]> على أسطر منفصلة حتى يكون العرض أكثر سهولة.

وهناك حرية اختيار لتجاهل أي إعلان أو أية مجموعة مركبة من الإعلانات أو العنـــاصر أو الكيانات أو السمات أو حتى كتل IGNORE أخرى، لكن يجب تجاهل إعلانات بأكملها. ويجب إن تتضمن شاشة IGNORE الإعلانات الكاملة التي تحذفها من DTD، واعلم أيضا أنـــه لا يمكنـــك تجاهل جزءاً من إعلان "مثل NDATA GIF في إعلان كيان غير محلل".

كما يمكن أيضا معرفة أن هناك مقطع إعلانات معين لم يتم تجاهله، ويكـــون بنـــاء الجملـــة بالنسبة لتعليمات INCLUDE تماماً كما هو مع تعليمات IGNORE لكن مـــع اختـــلاف الكلمـــة الأساسية.

```
<![ INCLUDE
  declarations that are included
]]>
```

عندما تكون INCLUDE داخـــل IGNORE فـــهذا يعنـــي أن INCLUDE فـــهذا يعنـــي أن IGNORE وإعلاناتها متجاهلة، وكذلك عندما يكون IGNORE داخل INCLUDE فــــهذا يعنـــي أيضاً أن إعلانات IGNORE متجاهلة ومن ثم يمكن التوصل إلى أن INCLUDE لا تبطل أبـــداً عمل IGNORE.

لذلك قد تثار التساؤلات حول سبب وجود INCLUDE، خاصة وأنه لن تتغير DTD لو أنه تـم إبعاد كل كتل INCLUDE وإبقاء محتوياتها فقط، فتبدو INCLUDE غير جوهرية بـــالمرة لكـــن هناك خاصية واحدة محكمة ذات مراجع معامل الكيان وIGNORE وINCLUED لا يمكنك فيـــها الاكتفاء بـــ IGNORE وحدها أولاً عرف مرجع معامل الكيان كالآتي:

```
<!ENTITY % fulldtd "IGNORE">
```

يمكنك تجاهل بعض العناصر بإخفائها في الشاشة التالية:

```
<![ %fulldtd;
  declarations
]]>
```

وعندما يكون مرجع معامل الكيان IGNORE %fulldtd، يتم تجاهل جميع الإعلانات، ولـــو أنك قمت بتحرير الكلمة الواحدة لتغير fulldtd من IGNORE إلى INCLUDE كالأتي:

```
<!ENTITY % fulldtd "INCLUDE">
```

فإنه على الفور ستتحول كل كتل IGNORE إلى كتل INCLUDE وفي الواقع سيكون لديـــك رمز تبديل على سطر واحد حتى يمكنك تشغيل الكتل أو التخلص منها.

وقد استخدمت في هذا المثال رمز تبديل واحد فقط وهو fulldtd والذي يمكن استخدامه مـــع كتل IGNORE/INCLUDE المتعددة الموجودة في DTDs، كما يمكـــن أيضـــاً أن يتـــم تشـــغيل المجموعات المختلفة لكتل IGNORE/INCLUDE أو التخلص منها، وذلك طبقاً لشروط المختلفة، وتظهر أهمية ذلك عند تصميم DTD لتضمينه في DTD آخر وقد يغير DTD الأساسي من شـكل DTD المضمن بتغيير قيمة رمز تبديل معامل الكيان.

خلاصة

لقد تعرفت في هذا الفصل على كيفية تضمين بيانات من غير نوع XML في مستندات XML وذلك من خلال الملاحظات والكيانات الخارجية غير المحللة والتعليمات المعالجة، وتعرفت تحديداً على تلك النقاط:

- توضح الملاحظات نوع البيانات من غير نوع XML.
- تعد الكيانات الخارجية غير المحللة وحدات تخزين تحتوي على نص من غير نوع XML أو بيانات ثنائية.
- يمكن تضمين الكيانات الخارجية غير المحللة في المستندات باستخدام إحدى سمتي ENTITY أو ENTITIES.
- تحتوي التعليمات المعالجة على تعليمات يقوم معالج XML بتمريرها كما هي إلى المستند الأساسي.
- تحدد كتل INCLUDE وIGNORE ما إذا كانت الإعلانات التي ينطوي عليها DTD يتم التعامل معها أو تجاهلها عند تحليل المستند.

وستتعرف من خلال الأجزاء القادمة من هذا الكتاب على أمثلة لمستندات مع DTD، لكن هذا الفصل يقدم المعلومات اللازمة عن البناء والاستخدام الأساسيان لاستكشاف DTD وفي الجزء الثالث سنبدأ الحديث عن أنماط لغات XML بدءاً بأوراق الأنماط المتتالية، المستوى الأول "Cascading Style Sheets, Level 1" في الفصل التالي.

٣

لغات الأنماط



أوراق نمط المستوى ١

CSS: هي لغة سهلة ومباشرة جداً لتطبيق الأنماط مثل النمط العريض وHelvetica لعناصر XML المعينة. ومعظم الأنماط التي تدعمها CSS هي أنماط شائعة أكثر من أنماط الكتابة الأخرى. ومثلا يمكنك اختيار الخط ونوعه وحجمه ولون الخلفية وكذلك المسافات بين العناصر المختلفة وكذلك الحدود حول العناصر وما هو أكثر من ذلك. ومع ذلك فبدل من أن تكون كل المعلومات الخاصة بالنمط مخزونة في المستند نفسه فإن المعلومات كلها تخزن في وثيقة منفصلة تسمى ورقة النمط. ويمكن تنسيق مستند XML واحد بعدة طرق مختلفة عن طريق تغيير ورقة النمط فقط. ويتم تصميم أوراق نمط متعددة للأغراض المختلفة: للطباعة أو للويب أو للاستخدامات الأخرى وكل منها تناسب الوسط المعين المخصص لها ودون تغيير محتوى المستند ذاته.

ما هي CSS

تستخدم CSS للإشارة إلى Cascading Style Sheets من الآن فصاعداً وقد عرفت في ١٩٩٦ على إنها وسائل قياسية لإضافة المعلومات الخاصة بخصائص النمط مثل الخط والحدود إلى مستندات HTML. غير أن CSS تعمل بصورة أفضل مع XML عنها مع HTML لأن HTML تعوقها الموائمة الخلفية بين علامات CSS وعلامات HTML. فمثلاً حتى يتم تدعيم خاصية CSS مثل *nowrap* يجب إزالة خاصية NOWRAP الغير قياسية ولكنها الأكثر استخداماً من HTML. ولأن عناصر XML ليس لها أي تنسيق مسبق فإنها لا تحدد نوع نمط CSS الذي يمكن إجراؤه على العناصر.

وورقة نمط CSS هي قائمة من القواعد وكل قاعدة تعطي أسماء العناصر التي تطبق عليها وكذلك الأنماط التي تود تطبيقها على هذه العناصر: فمثلاً فيما يتعلق بالقائمة ١٢-١ وهي ورقة نمط CSS للقصائد. فإن هذه الورقة تقدم خمسة قواعد. ويوجد لكل واحدة منها معامل اختيار- أسم العامل الذي ستطبق عليه- وكذلك قائمة بالخصائص التي سوف تطبق على العامل. والقاعدة الأولى لعامل POEM هي أنها يجب أن تعرض ككتلة قائمة بذاتها (display: block). والقاعدة الثانية هي أن محتويات عنصر العنوان TITLE يجب أن تعرض ككتلة قائمة بذاتها (display: block) وتتكون من ١٦ نقطة (font-size: 16pt) بخط عريض (font-weight: bold). أما القاعدة الثالثة هي أن عنصر POET يجب أن يعرض ككتلة قائمة بذاته (display: block) ويجب أن تبعد عما يليها بعشرة بكسل (margin-bottom: 10 px) والقاعدة الرابعة مثل سابقتها إلا أنه يتم تطبيقها على عناصر STANZA وأخيراً تحتم القاعدة الخامسة أن يتم عرض عناصر VERSE ككتلة قائمة بذاتها.

تعليمات البرمجة ١٢-١: ورقة نمط CSS للشعر

```
POEM   { display: block }
TITLE  { display: block; font-size: 16pt; font-weight: bold }
POET   { display: block; margin-bottom: 10px }
STANZA { display: block; margin-bottom: 10px }
VERSE  { display: block }
```

في عام ١٩٩٨ تم نشر حدود معدلة وموسعة لـ CSS وتـم تسـميتها CSS Level 2 أي CSS2 وفي ذات الوقت تم تسمية النسخة الأصلية من CSS Level 1 CSS أي (CSS1). وتعتبر CSS2 إعداد متقدم من CSS بخلاف بعض الاستثناءات البسيطة والتي سوف يتم الإشارة إليـها عند التعرض لها. وبعبارة أخرى تعتبر أوراق النمط CSS2 هي CSS1 مضاف إليـها الأنمـاط السمعية وأنواع الوسائط ومحدد الخصائص وخصائص أخرى جديدة. وبالتالي فـإن معظـم مـا سيقال في هذا الفصل ينطبق على كل من CSS1 وCSS2. وسيتم في الفصـل التـالي مناقشـة CSS2 على اعتبارها امتداد لـCSS1.

ويدعم كل من Netscape Navigator 4.0 وinternet Explorer 4.0 and 5.0 أجـزاء من CSS Level 1 ولكن لسوء الحظ ليست دائما نفس الأجزاء. ويفترض أن يقوم Mozilla 5.0 بتقديم دعماً متكافئاً لـCSS Level 1 ومعظم أجزاء CSS Level 2. ولكـن يقـوم Internet Explorer 5.0 بمهمة أفضل من Internet Explorer 4.0 في هذا الشأن ولكنها لا تزال تفتقـد بعض الأجزاء الحيوية وبخاصة فيما يتعلق بنـوع الصنـدوق وعنـاصر pseudo-elements وسوف أحاول الإشارة إلي المواقع التي قد تتسبب في مشاكل للمستعرض.

إضافة أوراق النمط إلى المستندات

حتى تفهم المقصود من ورقة النمط الموجودة في القائمة ١٢-١ يجب أن تلحق بها مسـتند XML لكي تعمل معها. وتعرض القائمة ١٢-٢ قصيدة من كتاب والت وايتمان للشعر الكلاسيكي أوراق الحشائش وتم تعيينها بـXML والسطر الثاني <?xml-stylesheet?> هي تعليمات لمسـتعرض الويب الذي يحمل هذا المستند لكي يطبق ورقة النمط الموجودة في ملـف poem.css الخـاص بهذا المستند ويوضح شكل ١٢-١ المستند المحمل بأسلوب ألفا القديم من Mozilla.

تعليمات البرمجة ١٢-٢: قصيدة Darest Thou Now O Soul مكتوبة بـ XML

```
<?xml version="1.0"?>

<?xml-stylesheet type="text/css" href="poem.css"?>

<POEM>

  <TITLE>Darest Thou Now O Soul</TITLE>
  <POET>Walt Whitman</POET>

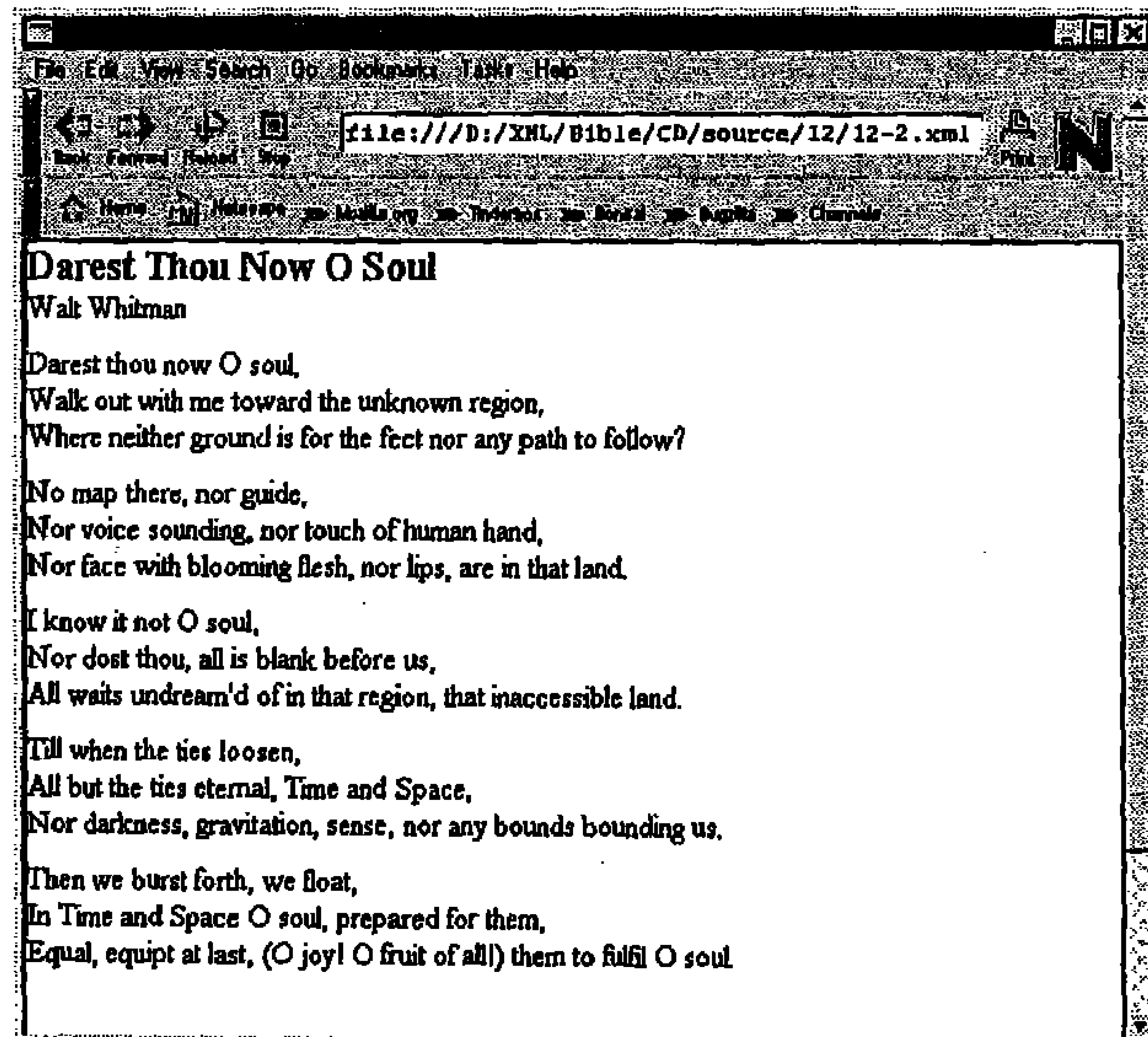
  <STANZA>
    <VERSE>Darest thou now O soul,</VERSE>
    <VERSE>Walk out with me toward the unknown region,</VERSE>
    <VERSE>Where neither ground is for the feet nor
any path to follow?</VERSE>
  </STANZA>
   <STANZA>
     <VERSE>No map there, nor guide,</VERSE>
     <VERSE>Nor voice sounding, nor touch of
           human hand,</VERSE>
     <VERSE>Nor face with blooming flesh, nor lips,
           are in that land.</VERSE>
   </STANZA>
   <STANZA>
     <VERSE>I know it not O soul,</VERSE>
     <VERSE>Nor dost thou, all is blank before us,</VERSE>
     <VERSE>All waits undream'd of in that region,
           that inaccessible land.</VERSE>
   </STANZA>
   <STANZA>
     <VERSE>Till when the ties loosen,</VERSE>
     <VERSE>All but the ties eternal, Time and Space,</VERSE>
     <VERSE>Nor darkness, gravitation, sense,
           nor any bounds bounding us.</VERSE>
```

```
</STANZA>
 <STANZA>
  <VERSE>Then we burst forth, we float,</VERSE>
  <VERSE>In Time and Space O soul,
       prepared for them,</VERSE>
  <VERSE>Equal, equipt at last, (O joy! O fruit of all!)
      them to fulfil O soul.</VERSE>
</STANZA>

</POEM>
```

وخاصية type الموجودة في تعليمات كتابة <?xml-stylesheet?> هي نوع MIME لورقة النمط التي تستخدمها وقيمتها هي text/css لـ CSS و text/xsl لـ XSL.

المرجع: يتم مناقشة CSS Level 2 في الفصل ١٣ وتتم مناقشة XSL فـي الفصليـن ١٤ و١٥.



الشكل ١٢-١ قصيدة Darest Thou Now O Soul مكتوبة بأسلوب Mozilla

وقيمة خاصية href في تعليمات عملية <?xml-stylesheet?> هي URL وهي ترتبط في أي مكان توجد فيه ورقة النمط. وإذا لم يتم التمكن من إيجاد ورقة النمط فإن مستعرض الويب غالباً سيستخدم النموذج الافتراضي الخاص بها ولكن قد تعرض بعض المستعرضات رسالة خطأ بدلاً من ذلك.

ويمكنك تطبيق نفس ورقة النمط للعديد من المستندات وغالباً ما ستفعل ذلك. وعليه فإنه من الطبيعي أن تضع أوراق النمط في موقع مركزي في موقع خادم الويب حيث تتمكن كل مستنداتك من الاستناد إليهم والموقع المناسب هو دليل الأنماط في المستوى الأول من خادم الشبكة:

```
<?xml-stylesheet type="text/css "href="/styles/poem.css"?>
```

ويمكنك أيضاً أن تستخدم URL المطلقة لورقة النمط في موقع ويب خارجي. ولكن هذا سيجعل موقعك معتمداً على حالة الموقع الخارجي.

```
<?xml-stylesheet type="text/css"
    href="http://metalab.unc.edu/xml/styles/poem.css"?>
```

بل ويمكنك أيضاً تكرار تعليمات <?xml-stylesheet?> لكي تسحب القواعد من أوراق نمط أخرى مثل:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="/styles/poem.css"?>
<?xml-stylesheet type="text/css"
    href="http://metalab.unc.edu/xml/styles/poem.css"?>
<POEM>
...
```

CSS وHTML مقابل CSS مع XML

بالرغم من أن هذا الكتاب يركز على أوراق نمط XML وCSS إلا أنه يتعامل أيضاً مع مستندات HTML والفارق الأساسي بين CSS مع HTML وCSS مع XML هو:

١- أن العناصر التي يمكنك أن تطبق عليها القاعدة تكون قاصرة فقط على عناصر HTML مثل P وPRE وLI وDIV وSPAN.

٢- لا تتعرف مستعرضات HTML على التعليمات وبالتالي فإن أوراق الأنماط ترتبط مع مستندات HTML وتستخدم تبويب LINK في عنصر HEAD. ويعد ذلك يمكن إضافة تعليمات خاصة بكل مستند على حدة في HEAD من عنصر STYLE مثل:

CSS وHTML مقابل CSS مع XML

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="/styles/poem.css" >
<STYLE TYPE="text/css">
  PRE { color: red }
</STYLE>
```

٣- لا توفر مستعرضات HTML خصائص CSS بنفس كفاءة مستعرضات XML وذلك بسبب أسلوب تنسيق العناصر. وتعتبر الجداول مشكلة كبيرة في هذا الشأن.



تعتبر أوراق النمط بطريقة أو بأخرى مستقلة عن DTDs. فالمستند المعتمد على ورقة النمط قد يحتوي أو لا يحتوي على DTD. والعكس صحيح أيضاً فالمستند المحتوي على DTD قد يوجد به أو لا يوجد به ورقة نمط. ومع ذلك لا تعمل DTDs دائماً كقائمة عملية للعناصر التي تود تزويدها بقواعد النمط.

وفي هذا الفصل والفصول التي تليه سوف تستخدم معظم الأمثلة مستندات منسقة جيداً ولكنها غير سارية المفعول. وعدم وجود DTDs سوف يجعل الأمثلة أقصر مما يزيد وضوح الأجزاء المقصودة. ولكن من الناحية العملية فإن معظم المستندات التي سوف ترفقها بأوراق النمط غالباً ما ستكون مستخدمة مع DTDs.

تحديد العناصر

تسمى أجزاء قاعدة CSS التي سيتم تطبيق العنصر عليها "المخصص a selector". وأكثر هذه المخصصات شيوعاً هو ببساطة أسم العنصر فمثلا TITLE في هذه القاعدة:

```
TITLE { display: block; font-size: 16pt; font-weight: bold }
```

ومع ذلك يمكن أن تستخدم هذه المخصصات مع عدة عناصر والعناصر ذات CLASS المعين أو خصائص ID والعناصر التي تظهر في سياق المرتبط بعناصر أخرى.



لا يمكنك عمله مع CSS Level 1 تحديد العناصر ذات أسماء أو قيم الخصائص المعينة خلاف ما تم تعريفه مسبقاً في خصائص CLASS أو ID ولكي تقوم بذلك عليك استخدام CSS Level 2 أو XSL.

تجميع المخصصات

إذا أردت إضافة مجموعة واحدة من الخصائص إلي عناصر متعددة يمكنك أن تضمن كل العناصر في المخصص مستخدماً الفواصل. وعلى سبيل المثال في قائمة ١٢-١ تمت صياغة كل من POET وSTANZA على أنها مجموعة متكاملة مكونة من ١٠ حافة بكسل. ويمكنك دمج القاعدتين كما يلي:

```
POET, STANZA { display: block; margin-bottom: 10px }
```

وبعد ذلك يمكن إضافة أكثر من قاعدة لعنصر محدد وبالتالي يمكنك دمج بعض الخصائص القياسية داخل قاعدة ذات مخصصات متعددة وبعدها أستخدم قواعد أكثر تخصصاً لكي تجري تنسيق مخصص للعناصر المحددة. فمثلاً في قائمة ١٢-١ تم وضع كل العناصر في القائمة للعرض ككتلة واحدة. ويمكن إجلاء ذلك في قاعدة واحدة بينما تحتوي قواعد منفصلة على التنسيقات الإضافية لكل من POET وSTANZA وTITLE كما يلي:

```
POEM, VERSE, TITLE, POET, STANZA { display: block }
POET, STANZA { margin-bottom: 10px }
TITLE {font-size: 16pt; font-weight: bold }
```

عناصر البيسيدو

تدعم CSS1 عنصري بيسيدو والتي يمكنها أن تتوجه إلي بعض الأجزاء من المستند والتي عادة ما يتم الإشارة إليها على أنها عناصر منفصلة العناصر والتي تحتاج إلي أنماط مختلفة. وهي السطر الأول والحرف الأول من العنصر.



الإصدارات الحديثة من Internet Explorer 5.0 وما يسبقها من Internet Explorer لا تدعم عناصر البيسيدو هذه. وكذا البيتا القديمة من Mozilla 5.0 وتدعمها فقط HTML.

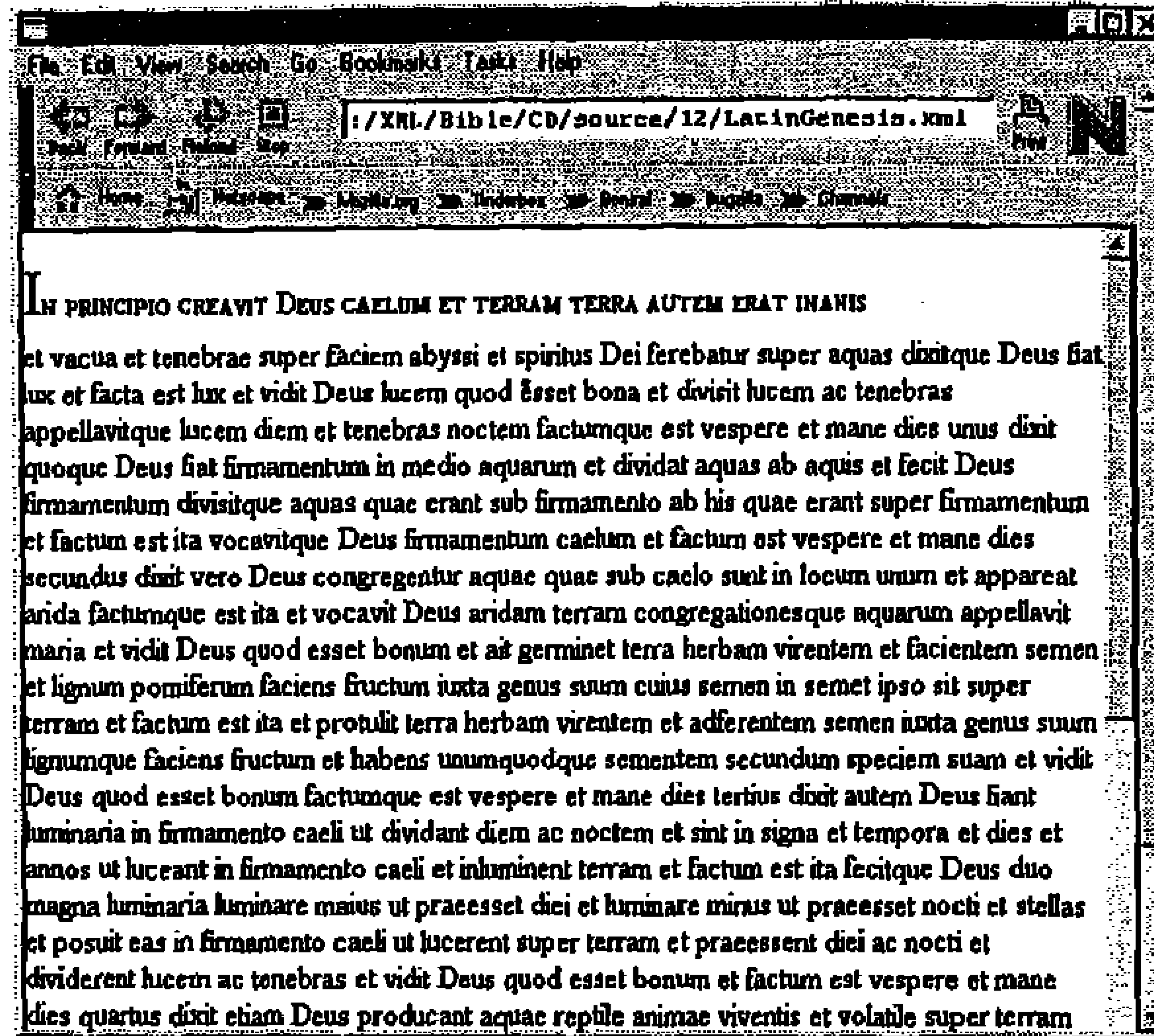
تخصيص الحرف الأول

من أهم أسباب تنسيق الحرف الأول من العنصر منفصلاً عن باقي العناصر هو إضافة إسقاط الأحرف الاستهلالية كما هو موضح في الشكل ١٢-٢. ويتم التوصل إلى ذلك عن طريق كتابة قاعدة يتم توجيهها للحرف الأول مع أسم العنصر ويتبعها الحرف الأول مثل:

```
CHAPTER:first-letter { font-size: 300%;
                       float: left; vertical-align: text-top }
```



كما تلاحظ في شكل ١٢-٢ فإن جزء "الإسقاط" من إسقاط الأحرف الاستهلالية (float: left; vertical-align: text-top) لا يبدو أنه بعمل مع البيتا القديمة من Mozilla 5.0 أو Internet Explorer 5.0 بالرغم من إمكانية تعديل حجم الحرف الأول.



الشكل ١٢-٢ إسقاط الأحرف الاستهلالية لعنصر البيسودو مع أحرف صغيرة تستخدم مع السطر الأول لعنصر للبيسودو

تخصيص السطر الأول

عادة ما يتم تنسيق السطر الأول من العنصر بطريقة مختلفة عن باقي النص للعنصر. فمثلاً بمكن كتابتها بحروف صغيرة بدلاً من نمط باقي المستند كما هو موضح في الشكل ١٢-٢ ويمكنك إضافة محدد first-line لأسم العنصر حتى تكون قاعدة يمكن تطبيقها على السطر الأول فقط مثل:

```
CHAPTER:first-line { font-variant: small-caps }
```

وما تحدده عناصر بيسودو هنا يتعلق بالتخطيط الخالي فإذا كان الإطار أكبر حجماً ويحتوي السطر الأول على كلمات أكثر ففي هذه الحالة ستكون الحروف بالحجم الصغير. .إذا أصبح الإطار أصغر حجماً أو أصبح حجم الخط أكبر فسيتغير التفاف النص وتقل عدد الكلمات

الموجودة في السطر الأول وعليه لن تصبح كلمات السطر الثاني بالحجم وتحديـــد أي الحــروف التي تشغل السطر الأول لعنصر البيسودو سيظل مختلفاً حتى يتم عرض المستند بالفعل.

فئات البيسودو

قد تحتاج أحياناً إلي استخدام نمط مختلف لعنصرين من نفس النوع فمثلاً أن يكون الخط أســـوداً عريضاً في أحدها وعادياً في الأخر. ولكي تتمكن من القيام بذلك أضف خاصية CLASS إلى أحد العناصر وبعدها أكتب قاعدة العنصر وفق CLASS المطلوبة.

ومثالاً على ذلك بيان بالكتب التي تحتوي على عدة CITATION وتوضح قائمة ١٢-٣ عينـة من ذلك. ولنفرض مثلاً أنك تود أن تلون كل المواقع الخاصة بألين ترننج باللون الأزرق وتـترك المواقع الأخرى كما هي. ولكي تقوم بذلك عليك أن تضيف خاصية CLASS بقيمة محددة. وتعمل TURING جيداً في تلوين العناصر.

تعليمات البرمجة ١٢-٣: بيان بالكتب بأسلوب XML تحتوي على ثلاثة عناصر CITATION

```
<?xml version="1.0" standalone="yes"?>
<?xml-stylesheet type="text/css" href="biblio.css:"?>
<BIBLIOGRAPHY>
 <CITATION CLASS="HOFSTADTER" ID="C1">
   <AUTHOR>Hofstadter, Douglas</AUTHOR>.
   "<TITLE>How Might Analogy, the Core of Human Thinking,
    Be Understood By Computers?</TITLE>"
   <JOURNAL>Scientific American</JOURNAL>,
   <MONTH>September</MONTH>
   <YEAR>1981</YEAR>
   <PAGES>18-30</PAGES>
 </CITATION>
 <CITATION CLASS="TURING" ID="C2">
  <AUTHOR>Turing, Alan M</AUTHOR>
  "<TITLE>On Computable Numbers,
```

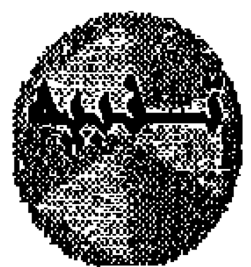


```
    With an Application to the Entscheidungs-problem</TITLE>"
   <JOURNAL>
    Proceedings of the London Mathematical Society</JOURNAL>,
   <SERIES>Series 2</SERIES>,
   <VOLUME>42</VOLUME>
   (<YEAR>1936</YEAR>):
   <PAGES>230-65</PAGES>.
  </CITATION>
  <CITATION CLASS="TURING" ID="C3">
   <AUTHOR>Turing, Alan M.</AUTHOR>
   "<TITLE>Computing Machinery &amp; Intelligence</TITLE>"
   <JOURNAL>Mind</JOURNAL>
   <VOLUME>59</VOLUME>
   (<MONTH>October</MONTH>
   <YEAR>1950</YEAR>):
   <PAGES>433-60</PAGES>
  </CITATION>
</BIBLIOGRAPHY>
```



أحد خواص CSS Level المقلقة هي أنها تجعل المستندات المختلطة ضرورية أكثر. وهناك الكثير من علامات الترقيم في القائمة ١٢-٣ والتي ليست جزئاً من المحتوى مثل الأقواس الموضوعة حول عنصر YEAR وعلامات الاقتباس حول TITLE. وهذه عناصر أساسية يجب أن تكون جزءاً من ورقة النمط. بينما تسمح CSS Level 2 بإدراج مزيد من النص مثل علامات الترقيم أن توضع قبل وبعد العناصر.

وتستخدم ورقة النمط في القائمة ١٢-٤ مخصص CLASS لتلون عناصر في فئة TURING باللون الأزرق.



وتدعم خصائص CLASS بـ IE5 ولكن لا تدعمها Mozilla مثل إصدار ميلستون ٣. وربما تدعم Mozilla خصائص CLASS عند حلول الوقت الذي ستصدر فيه رسمياً.

تعليمات البرمجة ١٢-٤: ورقة النمط التي تظهر العناصر من فئة TURING باللون الأزرق

```
BIBLIOGRAPHY { display: block }
CITATION.TURING { color: blue }
CITATION { display: block }
JOURNAL  { font-style: italic }
```



في المستند الساري المفعول يجب توضيح خاصية CLASS على أنها خاصية من خصائص العناصر ومثلاً فيما يلي DTD لبيان الكتب المسرود في القائمة ١٢-٣.

```
<!ELEMENT BIBLIOGRAPHY (CITATION*)>
  <!ATTLIST CITATION CLASS CDATA #IMPLIED>
  <!ATTLIST CITATION ID    ID    #REQUIRED>

  <!ELEMENT CITATION   ANY>
  <!ELEMENT AUTHOR    (#PCDATA)>
  <!ELEMENT TITLE     (#PCDATA)>
  <!ELEMENT JOURNAL   (#PCDATA)>
  <!ELEMENT MONTH     (#PCDATA)>
  <!ELEMENT YEAR      (#PCDATA)>
  <!ELEMENT SERIES    (#PCDATA)>
  <!ELEMENT VOLUME    (#PCDATA)>
  <!ELEMENT PAGES     (#PCDATA)>
```

وعموماً أنا لا انصح باستخدام هذه الطريقة. فعليك محاولة إضافة محددات العنصر الإضافية للمستند بدلاً من الاعتماد على خصائص CLASS. ومع ذلك قد تكون خصائص CLASS هامة عندما تكون المعلومات التي تحددها لا ترتبط بعنصر معين.

التحديد باستخدام ID

قد يحتاج العنصر المميز إلي أسلوب مميز من النمط أحياناً ولذلك تحتاج إلي قاعدة تطبق على ذلك العنصر بعينه. فمثلاً لنفرض أنك تود أن يظهر أحد العناصر في القائمة باللون الأسود الداكن

حتى تظهره بالمقارنة بالعناصر الأخرى ففي هذه الحالة يمكنك كتابة قاعدة يمكن تطبيقها على خاصية ID للعنصر والمخصص هو أسم العنصر يتبعه # ثم قيمة ID.

ومثالاً على ذلك قائمة ١٢-٥ وهي ورقة نمط تحدد عنصر CITATION من قائمة ١٢-٣ لبيان الكتب وبها ID C3 وتحدد هذا العنصر فقط بـاللون الأسود الداكن وتظهر عناصر CITATION باللون الافتراضي. وتظهر كل عناصر CITATION الأخرى بالطراز الجماعي بينما تظهر عناصر JOURNAL بالحروف المائلة.

تعليمات البرمجة ١٢-٥: ورقة نمط تظهر عناصر CITATION الخاصة بـ ID C3 باللون الأسود الداكن

```
BIBLIOGRAPHY { display: block }
CITATION#C3  { font-weight: bold }
CITATION     { display: block }
JOURNAL      { font-style: italic }
```



تدعم IE5 مخصصات ID وكذلك Mozilla لعناصر HTML ولكن ليس الحال كذلك بالنسبة لعناصر XML في إصدار ميلستون ٣. ربما تكون Mozilla مدعمة تدعيماً كاملاً لمخصصات ID وقت إصدارها.

مخصصات السياق

غالباً ما يعتمد تنسيق العنصر على العنصر الأساسي ويمكنك كتابة القواعد التي توجد في أسم العنصر الأصلي فقط. وحتى تتمكن من القيام بذلك أضف أسم العنصر الأصلي في مقدمة العنصر المطبق عليه النمط. فمثلاً عنصر CODE داخل عنصر PRE قد ينتج عنه Courier من ١٢ نقطة ولكن إذا كان السياق في المستند مكتوباً بـ 10 Courier نقط والقاعدة التالية تحقق هذا المطلوب:

```
BODY     { font-family: Times, serif; font-size: 10pt }
CODE     { font-family: Courier, monospaced; font-size: 10pt }
PRE      { font-size: 12pt }
PRE CODE { font-size: 12pt }
```

ويعتبر ذلك على أنه داخل عنصر BODY يكون حجم الخط ١٠ ويتغير الحجم إلى Courier ولكن ذو ١٠ نقاط أيضاً. ومع ذلك إذا تغير عنصر CODE داخل عنصر PRE فإن حجم الخط سيزيد إلى ١٢.

ويمكنك أن تزيد ذلك بأن تجعل الصفحة تشبه الصفحة الأصلية. وكل صفحة تتحدد من الصفحة الأصلية لها وهكذا.فمثلاً تقول القاعدة التالية أن عنصر NUMBER داخل عنصر YEAR ومن داخله عنصر DATE ويجب أن تكتب بخط أحادي:

```
DATE YEAR NUMBER { font-family: Courier, monospaced }
```

وفي الواقع غالباً لن تحتاج إلي مستوى التخصيص. وفي الأحوال التي ستحتاج إلـى ذلـك. ويمكنك دائماً إعادة كتابة ورق النمط بحيث تعتمد على الوراثة والتتالي والوحدات المتعلقة وتعتمد بشكل أقل على التنسيقات المحددة.

خصائص النمط

عند كتابة المستندات يدوياً فإنه من الطبيعي أنك تود إضافة نمط معين إلى أحد العناصر بـالتحديد دون تحرير ورقة النمط للمستند وفي الواقع يمكنك تجاوز بعض أوراق النمط القياسية الافتراضية للمستندات التي لا تستطيع تغييرها. ويمكنك القيام بذلك بإضافة خاصية STYLE إلـي العنصـر وقيمة هذه الخاصية موجودة في قائمة بها فواصل منقوطة لخصائص النمـط للعنصـر. فمثـلاً تستعمل CITATION خاصية STYLE حتى تبدو بلون أسود ثقيل:

```
<CITATION CLASS="TURING" ID="C3" STYLE="font-weight: bold">
  <AUTHOR>Turing, Alan M</AUTHOR>
  "<TITLE>Computing Machinery &amp; Intelligence</TITLE>"
  <JOURNAL>Mind</JOURNAL>
  <VOLUME>59</VOLUME>
  (<MONTH>October</MONTH>
  <YEAR>1950</YEAR>):
  <PAGES>433-60</PAGES>
</CITATION>
```

وإذا تعارضت خصائص النمط المحددة في STYLE مع الخصائص المعرفة في ورقة النمـط عندها تكون الأسبقية للخصائص المحددة في النمط.

حاول تجنب خصائص STYLE بقدر الإمكان. فسوف تصبح مستنداتك أنظف وأسـهل فـي التعديل إذا احتفظت بمعلومات النمط في أوراق نمط منفصلة ومع ذلك فإنه في بعـض الأوقـات تكون خصائص STYLE سريعة جداً وعملية بحيث لا يمكن تجاهلها.

ومرة أخرى إذا كنت تستخدم هذا الأسلوب في مستند ساري فعليك إعلان خصـائص STYLE في ATTLIST لوصف العنصر الذي تقوم بإجراء النمط عليه فمثلاً:

```
<!ELEMENT CITATION ANY>
<!ATTLIST CITATION CLASS CDATA #IMPLIED>
<!ATTLIST CITATION ID    ID    #REQUIRED>
<!ATTLIST CITATION STYLE CDATA #IMPLIED>
```



تدعم IE5 خصائص STYLE وكذلك Mozilla لعناصر HTML ولكن ليس عناصر XML في إصدار ميلستون ٣. ربما تكون Mozilla مدعمة تدعيماً كاملاً لخصائص STYLE وقت إصدارها.

توريث الخصائص

لا تتطلب CSS أن تكون القواعد محددة على وجه الخصوص لكل خاصية من خصائص العنصر في المستند فمثلاً إذا كانت هناك قاعدة لم تحدد حجم خط للعنصر فإن العنصر يرث حجم خط العنصر الأصلي. وإذا لم تكن هناك قاعدة تحدد لون العنصر فإن العنصر يرث لون العنصر الأصلي. وكذلك بالنسبة لخصائص CSS وفي الحقيقة فإن العناصر التي لا يتم توريثها هي الخلفية وخصائص المربع:

فمثلاً انظر لهذه الخصائص:

```
P     { font-weight: bold;
        font-size: 24pt;
        font-family: sans-serif}
BOOK  { font-style: italic; font-family: serif}
```

وكذلك انظر إلى جزئية XML هذه:

```
<P>
  Michael Willhoite's <BOOK>Daddy's Roommate</BOOK> is
  the #10 most frequently banned book in the U.S. in the 1990s.
</P>
```

وبالرغم من أنه لم يتم تحديد font-size وfont- weight لعنصر BOOK فإنه سيتم إظهاره بـ٢٤ نقطة داكنة لأنه تابع لعنصر P. وسوف تظهر أيضاً بخط مائل لأن ذلك محدد في قاعدته. ويرث BOOK كل من font-weight وfont-size للمستند الأصلي P وإذا ظهر عنصر BOOK في سياق أحد العناصر الأخرى فعندها سيرث font-weight وfont-size لذلك العنصر.

وتعتبر font-family أكثر مراوغة لأن كل من P وBOOK تظهر عناصر متضاربة لهذه الخاصية. ومن داخل عنصر BOOK فإن خاصية font-family التي أعلن عنها BOOK تكون لها الأسبقية. وخارج عنصر BOOK تستخدم P's font-family. وعلى ذلك فإن جملة Daddy's

Roommate تتم كتابتها بنوع خط serif بينما تتم كتابة جملـة most frequently banned book بالخط sans serif.

وغالباً ما ستحتاج إلى مستند ثانوي ليقوم بوراثة تنسيق المستند الأصلي وعليـــه فإنـــه مـــن الضروري تحديد خصائص أي عنصر تحديداً دقيقاً فمثلاً افترض أني قد حددت أن BOOK قد تم كتابتها بخط ١٢ نقطة كما يلي:

```
BOOK { font-style: italic; font-family: serif; font-size: 12pt}
```

عندها ستظهر كما هو موضح في الشكل ١٢-٣.

Michael Willhoite's *Daddy's Roommate* is the #10 most frequently banned book in the U.S. in the 1990s.

الشكل ١٢-٣ BOOK مكتوب بخط ١٢ نقطة.

ويمكنك تصويب ذلك عن طريق استعمال مخصص السياق لكي ينتقي عناصر BOOK مـــن داخل عناصر P ولكن الأسهل وراثة font-size.

وهناك طريقة واحدة لتجنب مثل هذه المشاكل مع التحكم في حجم كل عنصـــر وذلـــك عـــن طريق استخدام وحدات فردية مثل ems وex's بدلاً من الوحدات المطلقة مثـــل النقـــط والبيكـــا والبوصة والسنتيمترات والمليمترات. وem هو حجم الحرف m وex هي ارتفاع الحرف x فـــي حجم الخط الحالي. وإذا زاد حجم الخط يزداد حجم كل شيء أخر بقاس بالـــ ems وexs.

وهناك خيار أخر مماثل يتوفر لبعض الخصائص ويستخدم وحدات النســـبة المئويـــة فمثـــلاً توضح القاعدة التالية حجم الخط لعنصر FOOTNOTE_NUMBER وتحدده بـــ٨٠ في المائة من حجم خـــط العنصـــر الأساســـي وإذا زاد حجـــم الخـــط الأساســـي أو نقـــص يتغـــير حجـــم FOOTNOTE_NUMBER بالتالي:

```
FOOTNOTE_NUMBER { font-size: 80% }
```

وتتغير النسبة من خاصية إلى أخرى ففي خاصية vertical-align تكون النسبة هي ارتفـــاع خط الخنصر ذاته. ولكن في خاصية الحواف margin تكون النسبة من عرض العنصر.

المتتاليات

من الممكن أن تلحق أكثر من ورقة نمط للمستند فمثلاً يمكن أن يكون للمستعرض نمط افتراضي الذي يمكن إضافته للنمط الذي وضعه مصمم الصفحة. وفي هذه الحالة قد يكون هناك من القواعد التي يمكن تطبيقها على هذه العنصر وعلى ذلك فإنه من الضروري أن تحدد نظام تطبيق العناصر وتسمى هذه العملية بالتتالي ومن هنا جاء أسم تتالي أوراق النمط.

وهناك العديد من الطرق لإلحاق أوراق نمط CSS بمستند XML:

١- طريقة تعليمات <?xml-stylesheet?> والتي يمكن تضمينها في مستند XML.

٢- يمكن أن تستورد ورقة النمط ذاتها ورقة نمط أخرى باستخدام @import.

٣- يمكن للمستخدم أن يحدد ورقة نمط للمستند باستخدام آليات موجودة في المستعرض.

٤- يقدم المستعرض أنماط افتراضية لكافة الخصائص.

موجهات @import

قد تحتوي أوراق النمط على موجهات التي تحمل أوراق النمط المخزونة في الملفات الأخرى ويستخدم URL مطلق أو مرتبط لتحديد ورقة النمط مثل:

```
@import url(http://www.w3.org/basicstyles.css);
@import url(/styles/baseball.css);
```

ويجب أن تظهر الموجهات @import عند مقدمة ورقة النمط قبل أي قاعدة. وغالباً ما تتجاوز القواعد الموجودة في الأوراق الأصلية تلك القواعد الموجودة في الأوراق المستوردة. وتظهر أوراق النمط المستوردة بنفس النمط الذي تم استيرادها عليه. فلا يسمح بالدائرة مثل poem.css تقوم باستيراد stanza.css والتي تقوم باستيراد poem.css.

إعلان !important

تقوم قواعد CSS1 للكاتب بتجاوز قواعد القارئ إلا إذا كانت مرتبطة بإعلان !important للخاصية وتنص القاعدة التالية على أن عنصر TITLE يجب أن يكون ملوناً بلون أزرق حتى إذا كانت قاعدة الكاتب تنص على لون أخر. ومن ناحية أخرى فإن font-family.

يجب أن تكون serif في حالة عدم موافقة قاعدة الكاتب فقط.

```
TITLE { color: blue !important font-family: serif}
```

ومع ذلك يمكن إعلان قاعدة الكاتب أنها مهمة وفي هذه الحالة تتجاوز قاعدة الكاتب قاعدة القارئ.

ملحوظة

هذه فكرة سيئة للغاية حيث يجب أن يكون للقارئ دائماً الخيار لاختيار الطريقة التي يشاهدون بها شئ ما فمثلاً لا يمكن كتابة ورقة نمط بالألوان للقراء الذين يستخدمون المستعرض الأبيض والأسود ولا لذوي عاهات النظر ولا مستخدمي الشاشات ٢١ بوصة ولا مستعرضي أجهزة التلفزيون أو PDAs ويحدد العديد من مصممي الويب الأنماط التي يستخدمونها وبالتالي يخرجون صفحات يستحيل قراءتها إلا بواسطة أنظمة تماثل أنظمتهم تماماً ومن حسن الحظ أن CSS2 تقوم بعكس هذه القواعد لتلائم قاعدة القارئ.

نظام التتالي

يتم اختيار النمط من قواعد النمط للعنصر وعموماً تكون الأفضلية للقواعد الأكثر تخصصاً فمثلاً تأمل هذه الجزئية:

```
<OUEVRE>
  <PLAY ID="x02" CLASS="WILDE">
   The Importance of Being Earnest
  </PLAY>
</OUEVRE>
```

ويفضل استخدام القواعد الأكثر تحديداً وعلى ذلك فإنه يفضل استخدام القاعدة التي تحدد PLAY عن طريق ID الخاص به عن تلك التي تختار PLAY عن طريق CLASS. ويفضل القاعدة التي تحدد PLAY عن طريق CLASS عن تلك التي تحدد PLAY عن طريق عناصر OUEVRE. وأخيراً ففي حالة عدم تطبيق أي من القواعد الأخرى يتم تحديد قاعدة PLAY شاملة. وإذا لم يتوافق أي محدد يتم استخدام القيمة التي يتم توريثها من العنصر الأساسي. وإذا لم يكن هناك قيمة موروثة من العنصر الأساسي تستخدم القيمة الافتراضية.

وإذا كانت هناك أكثر من قاعدة في مستوى معين من التخصيص يتم استخدام نظام التتالي الذي يلي ذكره حسب نظام الأولوية التالية:

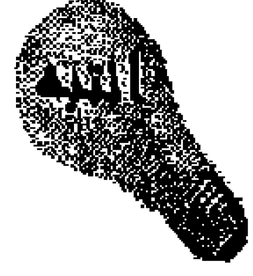
١- الإعلانات التي يحدد أهميتها الكاتب.

٢- الإعلانات التي يحدد أهميتها القارئ.

٣- الإعلانات التي لا يحدد أهميتها الكاتب.

٤- الإعلانات التي لا يحدد أهميتها القارئ.

٥- أخر قاعدة في ورقة النمط.



حاول تجنب نظام التتالي فإنه من الأصوب أن تحدد أقل عدد ممكن من النمط وأن تترك التحكم لمستعرض القارئ وللقارئ نفسه.

بعض التعليقات على أوراق النمط CSS

يمكن أن تتضمن أوراق النمط CSS على بعض التعليقات. وتعليقات CSS هي مثـــل C's /* */ وليست مثل تعليقات XML <!-- --> وتعليقات HTML. كما تعرض قائمة ١٢-٦. وورقة الرمـــز لا تطبق القواعد على العناصر تلقائياً بل هي تقوم أيضاً بـــالوصف بالإنجليزيـــة للنتـــائج التـــي يستهدف القيام بها.

تعليمات البرمجة ١٢-٦: أوراق النمط للقصائد ومعها التعليقات

```
/* Work around a Mozilla bug */
POEM { display:block }

/* Make the title look like an H1 header */
TITLE  { display: block; font-size: 16pt; font-weight: bold }
POET   { display: block; margin-bottom: 10 }

/* Put a blank line in-between stanzas,
   only a line break between verses */
STANZA { display: block; margin-bottom: 10 }
VERSE  { display: block }
```

وخصـــائص الالتفاف لــ CSS ليســـت مثـــل XML DTDs ولا Java أو C أو Perl ولكـــن التعليقات لا يجب أن تكون بلغة أخرى. ولكنها تقوم فقط بمساعدة الشخص الذي يحاول فهم ورقة النمط التي كتبتها ولا يستطيع سؤالك بشكل مباشر.

وحدات CSS

خصائص CSS لها أسماء وقيم ويسرد الجدول ١٢-١ بعض أسماء هـــذه الخصـــائص وبعـــض قيمها.

والأسماء هي الكلمات CSS الأساسية ولكن القيم تتعارض كثيراً. وبعضها كلمات أساسية مثل none في display و none أو solid في border-style: solid. والقيم الأخرى هـي أرقـام ذات وحدات مثل 0.5in أو 12pt في font-size: 12pt. والقيم الأخرى هي قيـم URL مثـل http://www.idgbooks.com/images/paper.gif فـي background-image: URL(http://www.idgbooks.com/images/paper.gif) أو ألوان RGB مثل #CC0033 in color: #CC0033. وتسمح الخصائص المختلفة بتطبيق قيم مختلفة ومع ذلك فهناك أربعـة أنواع فقط من القيم يمكن للخاصية أن تقوم إجراؤها. وهي:

١ – الطول.

٢ – URL.

٣ – اللون.

٤ – الكلمة الأساسية.

وتختلف الكلمات الأساسية من خاصية إلى أخرى ولكن القيم الأخرى لا تختلف من خاصيـة إلى أخرى وهذا يعني أن الطول يبقى كما هو أياً كانت الخاصية التي تتبعها هذه القيمة. فإذا كنت كيف تحدد طول الإطار فأنك بالتالي تعرف كيف تحدد طول الحافة أو الحشية أو الصورة . وهذا التكرار للتركيب يجعل التعامل مع الخصائص أكثر سهولة.

الجدول ١٢-١

نموذج لأسماء الخصائص وقيمها

Name	Value
Display	None
font-style	Italic
margin-top	0.5in
font-size	12pt
Border-style	Solid
Color	#CC0033
Background-color	White
Background-image	url(http://www.idgbooks.com/images/paper.gif)
list-style-image	url(/images/redbullet.png)
Line-height	120%

قيم الطول

يعتبر الطول في خاصية CSS نظام قياس يستخدم لقياس العرض والحجم وحجم الخـــط وكذلـــك مسافات الأحرف والكلمات وأحرف النص وكذلك ارتفاع الخط والحشو وعرض الإطار والعديـــد من الخصائص الأخرى ويمكن تحديدها بالثلاث وسائل التالية:

١- الوحدات المطلقة.

٢- الوحدات النسبية.

٣- وحدات الطول ذات النسب المئوية.

الوحدات المطلقة للطول

وحدات الطول المطلقة هي شئ غير حقيقي حيث لا يوجد بالفعل شئ أســـمه وحـــدات الطـــول المطلقة على شاشة الكمبيوتر. فتغيير درجة وضوح الشاشة من 640 إلى 480 إلى 1600 فـــي 1600 يغير كل ما يعرض على الشاشة بما في ذلك البوصات والسنتيمترات. وبالرغم من ذلـــك تدعم CSS خمسة وحدات مطلقة للطول والتي على الأقل لا تتغير من حجم خط إلى ما يليه ويتـم عرضها في جدول ١٢-٢.

الجدول ١٢-٢

وحدات الطول القياسية

Picas "pc"	Points "pt"	Millimeters "mm"	Centimeters "cm"	Inch "in"	
6	72	25.4	2.54	1.0	Inch
4.7244	28.3464	10	1	0.3937	Centimeters
0.47244	2.83464	1.0	0.1	0.03937	Millimeters
0.83333	1.0	0.352806	0.0352806	0.01389	Points
1.0	12	4.233	0.4233	0.16667	Picas

وتعرض الوحدات مع الأرقام المفترضة معها وأمامها وحدات الاختصار المســـتخدمة معـــها وهذه الوحدات هي:

البوصة in

السنتيمتر cm

المليمتر	mm
البوينتس	pt
البيكاس	PC

وقد يحتوي العدد على علامة عشرية مثل margin-top: 0.3in. وبعض الخصائص قد تعرض قيمة سلبية مثل 0.5in- ولكن لا تقوم جميعها بذلك. وحتى تلك التي تقوم بذلك تضع حدوداً على القيمة السالبة للطول. ومن الأفضل نجنب القيم السالبة التي تفوق أقصى عرض لموافقة المستعرض.

وحدات الطول النسبية

تدعم CSS أيضاً ثلاث وحدات نسبية للطول وهي:

١- em: وهي عرض الحرف m باستخدام عرض الخط الحالي.

٢- ex: وهي ارتفاع الحرف x باستخدام عرض الخط الحالي.

٣- px: وهي حجم البكسل "افترض البكسل المربعة حيث تستخدم كل العروض العامة الحديثة البكسل المربع بالرغم من أن بعض شاشات PC القديمة لا تستخدم ذلك".

وعلي سبيل المثال تقوم هذه القاعدة بتحديد الحدود اليمنى لعنصر PULLQUOTE بقيمة تساوي ضعف الحرف m باستخدام عرض الخط الحالي وفي الحدين العلوي والسفلي بارتفاع يساوي ارتفاع الحرف x باستخدام عرض الخط الحالي بمرة ونصف.

```
PULLQUOTE { border-right-width: 2em; border-left-width: 2em;
            border-top-width: 1.5ex; border-bottom-width: 1.5ex }
```

والغرض الطبيعي لاستخدام ems وex's هو تحديد العرض المناسب للخط المطلوب دون معرفة حجم الخط بالضرورة. فمثلاً في القاعدة العلوية حجم الخط غير معروف وبالتالي فأن عرض الإطار غير معروف أيضاً ويمكن أن يتحدد وقت العرض بالمقارنة مع m وx لحجم الخط المستخدم حالياً.

وأحجام الخط الكبيرة سيكون لها ems وex's أكبر حجماً. والأطوال باستخدام البكسل هي نسبية للارتفاعات وعروض البكسل على الشاشة. "المفترض استخدام البكسل المربع بها" وعروض وارتفاعات الصور يتم تحديدها بالبكسل عادةً.



استخدام وحدات القياس بالبكسل فكرة غير جيدة أولاً لأن حجم البكسل يختلف كثيراً مع ودرجات الوضوح ومعظم الأجهزة يتم أعدادها بدرجات عالية من الوضوح مما يجعل البكسل صغير جداً للاستجابة لها. وثانياً ستصبح الشاشات ٢٠٠ و٣٠٠ dpi شائعة الاستخدام في العشرة سنوات التالية. وأخيراً سيتم البعد عن استخدام وحدات القياس الـ٢٧ و٢٨ باكسيل لكل بوصة التي كانت سائدة منذ استخدام أول إصدار ماكينتوش عام ١٩٨٤. والمستندات التي تم تحديد مقاييسها باستخدام وحدات غير وحدات الشاشة مثل ems وex's والنقاط والبيكاس والبوصات ستتمكن من الانتقال. ولكن المستندات التي تعتمد على وحدات قياس البكسل ستكون صغيرة جداً عند رؤيتها على الشاشات ذات درجة الوضوح العالية.

وحدات الطول ذات النسب المئوية

وأخيراً يمكن تحديد الأدوات باستخدام النسب المئوية وعموماً هي نسبة مئوية للقيمة الحالية للخاصية. فمثلاً بالنسبة لعنصر STANZA هي ١٢ نقطة وحجم الخط الذي يحتوي على كل من VERSE وSTANZA يحتوي على ١٥٠ في المائة وبالتالي فإن حجم الخط لـ VERSE سوف يصبح ١٨ نقطة.

قيم URL

هناك ثلاثة من خصائص CSS يمكن أن يكون لها قيم URL وهي: background-image وlist-style-image وخاصية الاختصار لـ list-style. بالإضافة إلى قاعدة @import التي تستخدم قيم URL كما علمت من قبل. وقيم URL الحرفية توضع في url() وكل قيم URL المطلقة والنسبية مسموح بها. مثل:

```
DOC { background-image: url (http://www.mysite.com/bg.gif)
LETTER { background-image: url(/images/paper.gif) }
SOFTWARE { background-image: url(../images/screenshot.gif)}
GAME { background-image: url(currentposition.gif)}
```

ويمكنك أن تضع URL في علامتي ترقيم ثنائية أو أحادية بالرغم من أنه ليست هناك فائدة من ذلك مثل:

```
DOC { background-image: url(http://www.mysite.com/bg.gif) }
LETTER { background-image: url(/images/paper.gif) }
SOFTWARE { background-image: url(../images/screenshot.gif)}
GAME { background-image: url(currentposition.gif)}
```

ويجب فصل كل من الأقواس والفواصل والفواصل المنقوطة والنقطتين التي تظهر في URL بشرطة مائلة/ ويجب استبدال كل من الفواصل والأقواس والمسافات وعلامات الاقتباس التي تظهر في URL "وهي غير مألوفة باستثناء علامة المسافة" يجب استبدالها جميعاً بعلامة URL القياسية % كما يلي:

space	%20
,	%2C
'	%27
"	%22
(	%2B
)	%2C

ملحوظة

تقوم CSS بتحديد المسافات التي تترك لعلامات الخطوط المائلة العكسية الخاصة بها (\", \', \,, \(, \)) وإن كان ذلك يزيد إمكانية حدوث الخلط.

قيم الألوان

من أكثر استخدامات CSS شيوعاً والتي تفضلها عن HTML التقليدية هي قابليتها على استخدام الألوان في الجزء الأمامي وفي الخلفية لكل عناصر الصفحة تقريباً. والعناصر التي لها قيم لونية تشمل color و background-color و border-color. وتقدم CSS أربعة أساليب لتحديد اللون عن طريق الاسم أو عن طريق المكونات السدس عشرية أو عن طريق المكونات الصحيحة أو النسب المئوية. وتتعرف CSS على أسماء الـ ١٦ لوناً التالية التي نقلتها من مجموعة ألوان ويندوز VGA:

- ♦ النيلي Aqua
- ♦ الأسود Black
- ♦ الأزرق Blue
- ♦ الأرجواني Fuchsi
- ♦ الرمادي Gray
- ♦ الأخضر Green
- ♦ الليموني Lime
- ♦ البني Maroon
- ♦ الأزرق الداكن Navy
- ♦ الزيتوني Olive
- ♦ البنفسجي Purple
- ♦ الأحمر Red
- ♦ الفضي Silver
- ♦ البيج Teal
- ♦ الأبيض White
- ♦ الأصفر Yellow

وبالطبع تستطيع الشاشة الملونة أم تعرض ملايين الألوان أكثر من هـــذه. ويمكـــن تحديدهـــا بإضافة قيم مختلفة لمكونات ألوان الأحمر والأخضر والأزرق (RGB). وحيث أن CSS تفـترض وجود ٢٤ بت لنموذج اللون فإن كل من هذه الألوان الأساسية يتحدد بثماني بت. ومكون الثمـــاني بت هو مكون يقع بين الـــ٠ و٢٥٥. وقد يعطى هذا الرقم إما بالرقم العشري لـــ RGB أو السدس عشري لـــ RGB ويمكن أن تعطى أيضاً في صورة نسبة مئوية كبديل عن ذلك وتتراوح ما بيـــن (0) 0% و(255) 100%. وجدول ١٢-٣ يقدم بعض الألوان الممكنة والأرقـــام العشـــرية أو السدس عشرية والنسب المئوية لـــ RGBs.

الجدول ١٢-٣

عينات ألوان CSS

Color	Decimal RGB	Hexadecimal	RGBPercentage RGB
أحمر صافي	rgb(255,0,0)	#FF0000	Rgb(100%, 0%, 0%)
أزرق صافي	Rgb(0,0,255)	#0000FF	rgb(0%, 0%, 0%)
أخضر صافي	Rgb(0,255,0)	#00FF00	rgb(0%, 100%, 0%)
أبيض	Rgb(255,255,255)	#FFFFFF	rgb(100%,100%,100 %)
أسود	Rgb(0,0,0)	#000000	rgb(0%, 0%, 0%)
بنفسجي خفيف	Rgb(255,204,255)	#FFCCFF	rgb(100%, 80%, 100%)
رمادي متوسط	rgb(153,153,153)	#999999	rgb(60%, 60%, 60%)
بني	rgb(153,102,51)	#996633	rgb(60%, 40%, 20%)
وردي	rgb(255,204,204)	#FFCCCC	rgb(100%, 80%, 80%)
برتقالي	rgb(255,204,204)	#FFCC00	rgb(100%, 80%, 80%)



لا يزال الكثيرين يستخدمون شاشات تعرض ٢٥٦ لوناً بل إن بعض الألوان تبـــدو مختلفة تماماً على Macs وPCs وأكثر الألوان استخداماً هي الـــ١٦ لوناً الســـالف

ذكرها. ويليها تلك التي تتكون باستخدام المكونات السدس عشـوية ٠٠ و٣٣ و٦٦ و٩٩ وCC وFF "٠ و٥١ و١٠٢ و١٥٣ و٢٠٤ و٢٥٥" بالكسر العشري و"0%, 20%, 40%, 60%, 80% 100%" بوحدات النسب المئوية. مثـــلاً 33FFCC هو لون أمن للاستخدام على الشاشة لأن مكونات اللون الأحمر تتألف مـــن رقـــم ثلاثة مكرر مرتين ثم حرف F مكرر مرتين رمزاً للون الأخضر واللـــون الأزرق المكون من حرف C مرتين. وإذا حددت كسراً عشرياً للون RGB باستخدام ثلاثـــة أرقام فقط فأن CSS سوف تضاعفها. مثلاً FC0# هو فـــي الحقيقـــة FFCC00# وبالمثل 963# هو 996633#.

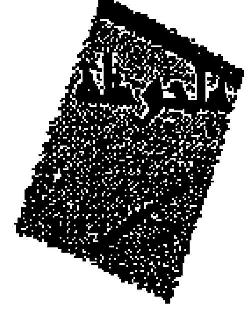
قيمة الكلمة الأساسية

الكلمة الأساسية هي أكثر القيم تغيراً من قيم خصائص CSS. وهي متغيرة مـــن خاصيـــة إلـــي أخرى. ولكن الخصائص المتماثلة عادة تدعم كلمة أساسية مماثلة. فمثلاً يمكن أن تكـــون قيمـــة border-left-style أي كلمة من الكلمات none أو dotted أو dashed أو solid أو double أو groove أو border-top-style أو border-bottom-style ويمكن لخصـــائص -border style أن تكون لها أحد القيم التاليـــة: none وdotted وdashed وsolid وdouble وgroove وridge وinset أو outset. والكلمات الأساسية سيتم دراستها على حدة في جزء كل خاصيـــة منفصلة.

تجميع وصف وإدراج العناصر في قوائم

من منطلق CSS Level 1 فإن كل العناصر إما تعتبر ذات مستوي تجمع أو مستوى الصـــف أو مدرجة في قائمة أو عناصر غير مرئية. "بينما تضيف CSS Level 2 مزيد مـــن الاحتمـــالات" ونوع العنصر المطلوب تحدده خصائص عرضه ولها أربعة قيم محتملة تعـــبر عنـــها الكلمـــات الأساسية:

block
inline
list-item
none



والقيمة الافتراضية لخاصية display في CSS Level 1 هي block وهذا يعني أن العنصر يظهر في المربع الخاص به ولكنه منفصل عن العناصر الأخرى بطريقة معينة. ومع ذلك يتغير الافتراض في CSS Level 2 إلي inline والذي يعني أن محتويات العنصر توضع بالتالي في النص بعد العنصر. وتستخدم معظم مستعرضات الويب الافتراض CSS 2 "الصف" بدلاً من CSS 1 "التجمع".

وتعتبر كل من عناصر HTML وEM وSTRONG وB وI وكذلك A من عناصر الصف. وفي مثال أخر يمكنك أن تفكر أن كل من EM وSTRONG وB وI وكذلك A هي رموز عنصر الصف. وهي ليست منفصلة عن باقي النص.

وتنفصل عناصر مستوى التجمع عن عناصر التجمع الأخرى عادة عن طريق خط فاصل. وتعتبر كل من HTML P وBLOCKQUOTE ومن H1 إلى H6 وHR هي أمثلة من عناصر مستوى التجمع. ويمكن أن تحتوي عناصر التجمع على عناصر صف وعناصر تجمع أخرى ولكن لا يجب أم تحتوي عناصر الصف على عناصر صف أخرى ولا عناصر تجمع أيضاً.

وعناصر الإدراج في القوائم هي عناصر من مستوى التجمع ويسبقها مكون لقوائم العنصر. وستتم مناقشة عناصر القوائم بالتفصيل في القسم التالي.

وأخيراً العناصر ذات خصائص display التي يتم أعدادها على none تكون غير مرئية على الشاشة. وهي لا تؤثر على وضع العناصر الأخرى المرئية في الصفحة. وفي حالة HTML توجد خصائص عرض لكل من TITLE وMETA وHEAD لعنصر none. أما في XML فإن display: none غالباً ما تكون مفيدة للمعلومات الغير مرئية للعنصر.

وتحتوي القائمة ١٢-٧ على خلاصة مسرحية وليم شكسبير الليلة الثانية عشر وتحتوي على العناصر التالية:

SYNOPSIS	ACT_NUMBER
TITLE	SCENE_NUMBER
ACT	LOCATION
SCENE	CHARACTER

ويمكنك القيام بمهمة كبيرة عند تنسيق هذه البيانات مستخدماً خصائص العرض SYNOPSIS وTITLE وACT وSCENE فقط. وهي كلها من عناصر مستوى التجمع. أما عن عناصر ACT_NUMBER وSCENE_NUMBER وLOCATION وCHARACTER فيمكن أن تبقى من عناصر الصف والقائمة ١٢-٨ هي ورقة رمز بسيطة يمكن أن تقوم بهذا الغرض:

تعليمات البرمجة ١٢-٧: خلاصة مسرحية وليم شكسبير الليلة الثانية عشر مكتوبة بـ XML

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="12-8.css"?>
<SYNOPSIS>
 <TITLE>Twelfth Night</TITLE>

 <ACT>
  <ACT_NUMBER>Act 1</ACT_NUMBER>
  <SCENE>
   <SCENE_NUMBER>Scene 1</SCENE_NUMBER>
   <LOCATION><CHARACTER>Duke Orsino</CHARACTER>'s palace
   </LOCATION>
  </SCENE>
  <SCENE>
   <SCENE_NUMBER>Scene 2</SCENE_NUMBER>
   <LOCATION>The sea-coast</LOCATION>
  </SCENE>
  <SCENE>
   <SCENE_NUMBER>Scene 3</SCENE_NUMBER>
   <LOCATION><CHARACTER>Olivia</CHARACTER>'s house
   </LOCATION>
  </SCENE>
  <SCENE>
   <SCENE_NUMBER>Scene 4</SCENE_NUMBER>
   <LOCATION><CHARACTER>Duke Orsino</CHARACTER>'s palace.
   </LOCATION>
  </SCENE>
  <SCENE>
   <SCENE_NUMBER>Scene 5</SCENE_NUMBER>
   <LOCATION><CHARACTER>Olivia</CHARACTER>'s house
```

```
      </LOCATION>
    </SCENE>
  </ACT>

  <ACT>
    <ACT_NUMBER>Act 2</ACT_NUMBER>
    <SCENE>
      <SCENE_NUMBER>Scene 1</SCENE_NUMBER>
      <LOCATION>The sea-coast</LOCATION>
    </SCENE>
    <SCENE>
      <SCENE_NUMBER>Scene 2</SCENE_NUMBER>
      <LOCATION>A street</LOCATION>
    </SCENE>
<SCENE>
      <SCENE_NUMBER>Scene 3</SCENE_NUMBER>
      <LOCATION><CHARACTER>Olivia</CHARACTER>'s house
      </LOCATION>
    </SCENE>
    <SCENE>
      <SCENE_NUMBER>Scene 4</SCENE_NUMBER>
      <LOCATION><CHARACTER>Duke Orsino</CHARACTER>'s palace.
      </LOCATION>
    </SCENE>
    <SCENE>
      <SCENE_NUMBER>Scene 5</SCENE_NUMBER>
      <LOCATION><CHARACTER>Olivia</CHARACTER>'s garden
      </LOCATION>
    </SCENE>
  </ACT>

  <ACT>
    <ACT_NUMBER>Act 3</ACT_NUMBER>
```

```
 <SCENE>
  <SCENE_NUMBER>Scene 1</SCENE_NUMBER>
  <LOCATION><CHARACTER>Olivia</CHARACTER>'s garden
  </LOCATION>
 </SCENE>
 <SCENE>
  <SCENE_NUMBER>Scene 2</SCENE_NUMBER>
  <LOCATION><CHARACTER>Olivia</CHARACTER>'s house
  </LOCATION>
 </SCENE>
 <SCENE>
  <SCENE_NUMBER>Scene 3</SCENE_NUMBER>
  <LOCATION>A street</LOCATION>
 </SCENE>
 <SCENE>
  <SCENE_NUMBER>Scene 4</SCENE_NUMBER>
  <LOCATION><CHARACTER>Olivia</CHARACTER>'s garden
  </LOCATION>
 </SCENE>
</ACT>

<ACT>
 <ACT_NUMBER>Act 4</ACT_NUMBER>
 <SCENE>
  <SCENE_NUMBER>Scene 1</SCENE_NUMBER>
  <LOCATION><CHARACTER>Olivia</CHARACTER>'s front yard
  </LOCATION>
 </SCENE>
 <SCENE>
  <SCENE_NUMBER>Scene 2</SCENE_NUMBER>
  <LOCATION><CHARACTER>Olivia</CHARACTER>'s house
  </LOCATION>
```

```
    </SCENE>
    <SCENE>
      <SCENE_NUMBER>Scene 3</SCENE_NUMBER>
      <LOCATION><CHARACTER>Olivia</CHARACTER>'s garden
      </LOCATION>
    </SCENE>
  </ACT>

  <ACT>
    <ACT_NUMBER>Act 5</ACT_NUMBER>
    <SCENE>
      <SCENE_NUMBER>Scene 1</SCENE_NUMBER>
      <LOCATION><CHARACTER>Olivia</CHARACTER>'s front yard
      </LOCATION>
    </SCENE>
  </ACT>

</SYNOPSIS>
```

تعليمات البرمجة ١٢-٨: هي ورقة رمز بسيطة لملخص المسرحية

```
SYNOPSIS, TITLE, ACT, SCENE { display: block }
```

ويوضح شكل ١٢-٤ ملخص مسرحية الليلة الثانية عشر موضوع في مزيلا لورقة رمز القائمة ١٢-٨. ولاحظ أنه في القائمة ١٢-٨ ليس من الضروري أن تحدد كل من ACT_NUMBER وACT_NUMBER وLOCATION وCHARACTER هي من عناصر الصف. وهذا هو الافتراض إلا إذا تم تحديد غير ذلك. وخاصية display لا يتم توريثها. وعلى ذلك فإن مجرد كون SCENE من عناصر التجمع لا يعني بالقطع أن توابعها SCENE_NUMBER وLOCATION هي الأخرى من عناصر مستوى التجمع.

عناصر القوائم

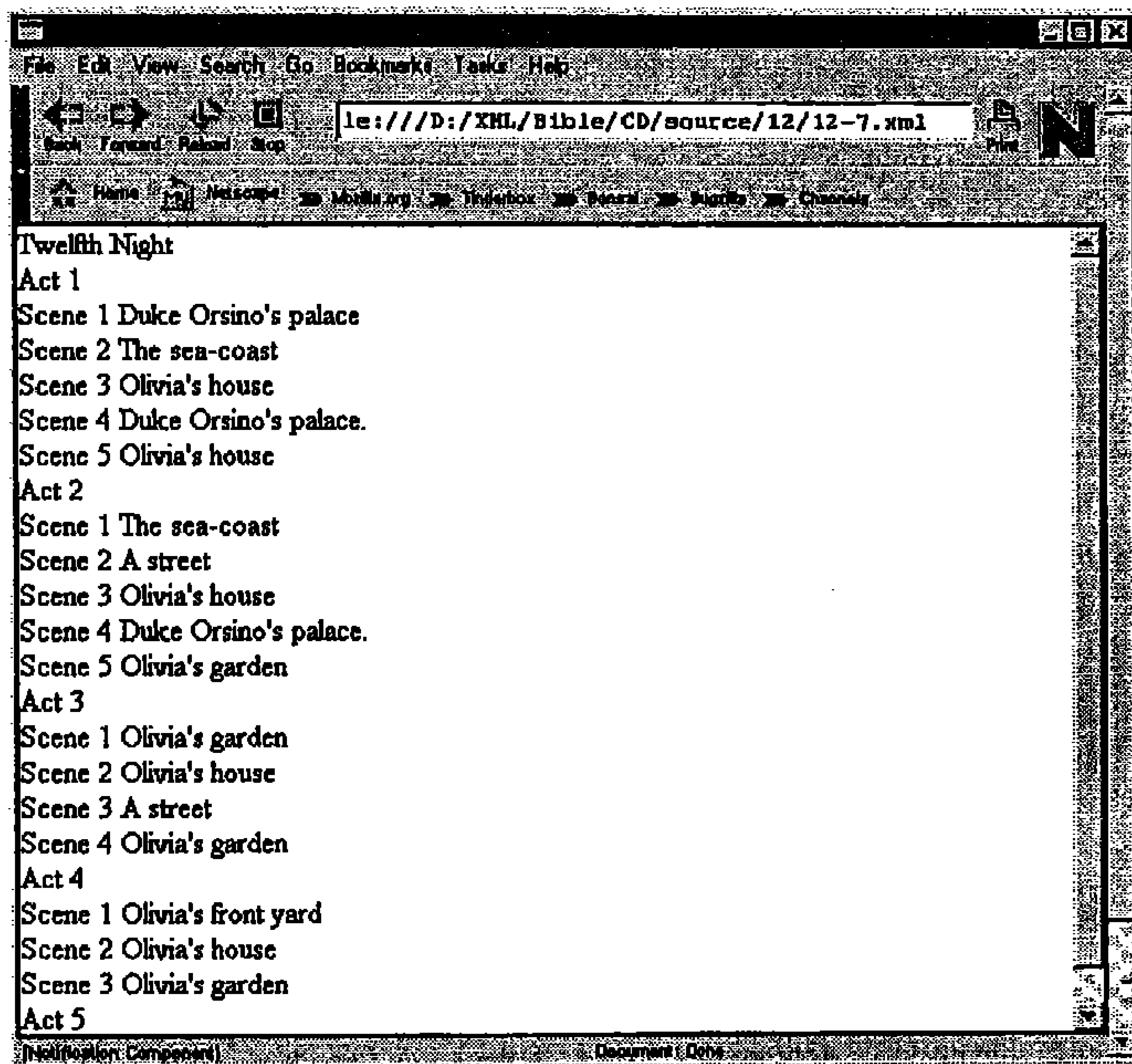
إذا اخترت أن تكون قيمة list-item لخاصية display هناك ثلاث خصـــائص إضافيـة عليـك إعدادها. وهذه الخصائص تؤثر على كيفية عرض عناصر القوائم وهي:

1- list-style-type
2- list-style-image
3- list-style-position

وهناك أيضاً خاصية list-style مختصرة تمكنك من إعداد الثلاثة داخل قاعدة واحدة.



لا تدعم كل مـن Explorer 5.0 وMozilla 5.0 milestone 3 حتـى الآن display: list-item. وتعاملها Mozilla على أنها عناصر تجمع بسـيطة بينمـا تعاملها Internet Explorer بما هو أسوء من ذلك فتعتبرها عناصر صف.



الشكل ١٢-٤ ملخص مسرحية الليلة الثانيـة عشـر فـي Mozilla 5.0.

خاصية list-style-type

تحدد خاصية list-style-type طبيعة النقطة الموجودة أمام كل عنصر قائمة الاحتمـالات كمـا يلي:

disc
circle
square
decimal
lower-roman
upper-roman
lower-alpha
upper-alpha
none

والافتراض هو disc مثل ورقة الرمز في القائمة ١٢-٩ والتي تطبق على الملخص الموجود في القائمة ١٢-٧ والتي تحدد كل من ACT وSCENE على أنها عناصر قائمة. ومع ذلـــك فـــلا توجد نقطة أمام ACT وتوجد نقطة مربعة أمام SCENE.

تعليمات البرمجة ١٢-٩: ورقة رمز ها ملخص المسرحية باستخدام عناصر القائمة

```
SYNOPSIS, TITLE { display: block }
ACT { display: list-item; list-style-type: none }
SCENE { display: list-item; list-style-type: square }
```

خاصية list-style-image

يمكنك استخدام صورة ذات نقط مضيئة حسب اختيارك وتحملها من الملف كنقطة. ولكـــي تقـــوم بذلك حدد خاصية list-style-image على URL للصورة. وعند إعداد كـــل مـــن list-style-image وlist-style-image فإن عنصر القائمة سوف يسبق بكل من الصورة وحرف النقطـــة. ومع ذلك فإنها حالة نادرة. وتستخدم القائمة ١٢-١٠ الرمز ♥ المخزون فـــي ملـــف heart.gif على أنه النقطة قبل كل مشهد "ومسرحية الليلة الثانية عشر مسرحية كوميدية شعرية علـــى أيـــة حال".

تعليمات البرمجة ١٢-١٠: ورقة الرمز لملخص المسرحية التي تستخدم خاصية style-image

```
SYNOPSIS, TITLE { display: block }
ACT { display: list-item; list-style-type: none }
SCENE { display: list-item;
    List-style-image: url(heart.gif); list-style-type: none }
```

خاصية list-style-position

تحدد خاصية list-style-position ما إذا كانت النقطة سيتم رسمها outside أو inside النص الذي يحتوي على عناصر القائمة. والقيم القانونية في outside وinside. أما القيم الافتراضية ففي outside فقط. والفارق يكون ملحوظاً فقط في حالة التفاف النص لأكثر من سطر. وفي حالة ما إذا كانت في الداخل ستكون كما يلي:

- If music be the food of love, play on/Give me excess of it, that, surfeiting,/The appetite may sicken, and so die./That strain again! it had a dying fall:

أما في حالة الخارج فهي كما يلي:

- If music be the food of love, play on/Give me excess of it, that, surfeiting,/The appetite may sicken, and so die./That strain again! it had a dying fall:

خاصية list-style Shorthand

وأخيراً خاصية list-style Shorthand وهي نوع من الاختصار يسمح لك أن تقوم بإعداد كل الخصائص الثلاثة السابقة معاً. مثلاً القاعدة التالية تعبر عن أنه سيتم عرض SCENE في الداخل مع وجود صورة القلب وبدون نقطة عند البداية:

```
SCENE { display: list-item;
        list-style: none inside url(heart.gif) }
```

خاصية whitespace

تحدد خاصية whitespace كيف ستبدو المسافة البيضاء مع العنصر "المسافات والتبويبات والخطوط الفاصلة". والقيم المسموح بها هي:

normal
pre
nowrap

والقيمة الافتراضية normal تعني ببساطة أن المسافة البيضاء تعتبر مسافة واحدة والكلمات يتم التفافها حتى تملئ مساحة الصفحة أو الشاشة، وهذه هي المعاملة الطبيعية للمسافات البيضاء في كل من HTML وXML.

وتعمل القيمة السابقة PRE تعمل عمل العنصر السابق تنسيقه في HTML. وكـــل المســـافات البيضاء في المستند المدرج تعتبر مخرجات يتم تكوينها بنفس الدقة في إخـــراج المســـتند. وقـــد يصحب ذلك تغييراً إلى حجم الخط ذو المسافات الأحادية وقد يكون ذلك مفيداً في حالة المصـــدر الذي يعتمد على الكمبيوتر أو في حالة الشعر. وفي القائمة ١٢-١١ قصيدة The Altar للشـــاعر جورج هربرت وفيها تعتبر المسافات هامة وتعتبر السطور ذات أهمية بالنسبة لموضوع القصيدة:

تعليمات البرمجة ١٢-١١ : *The Altar in XML*

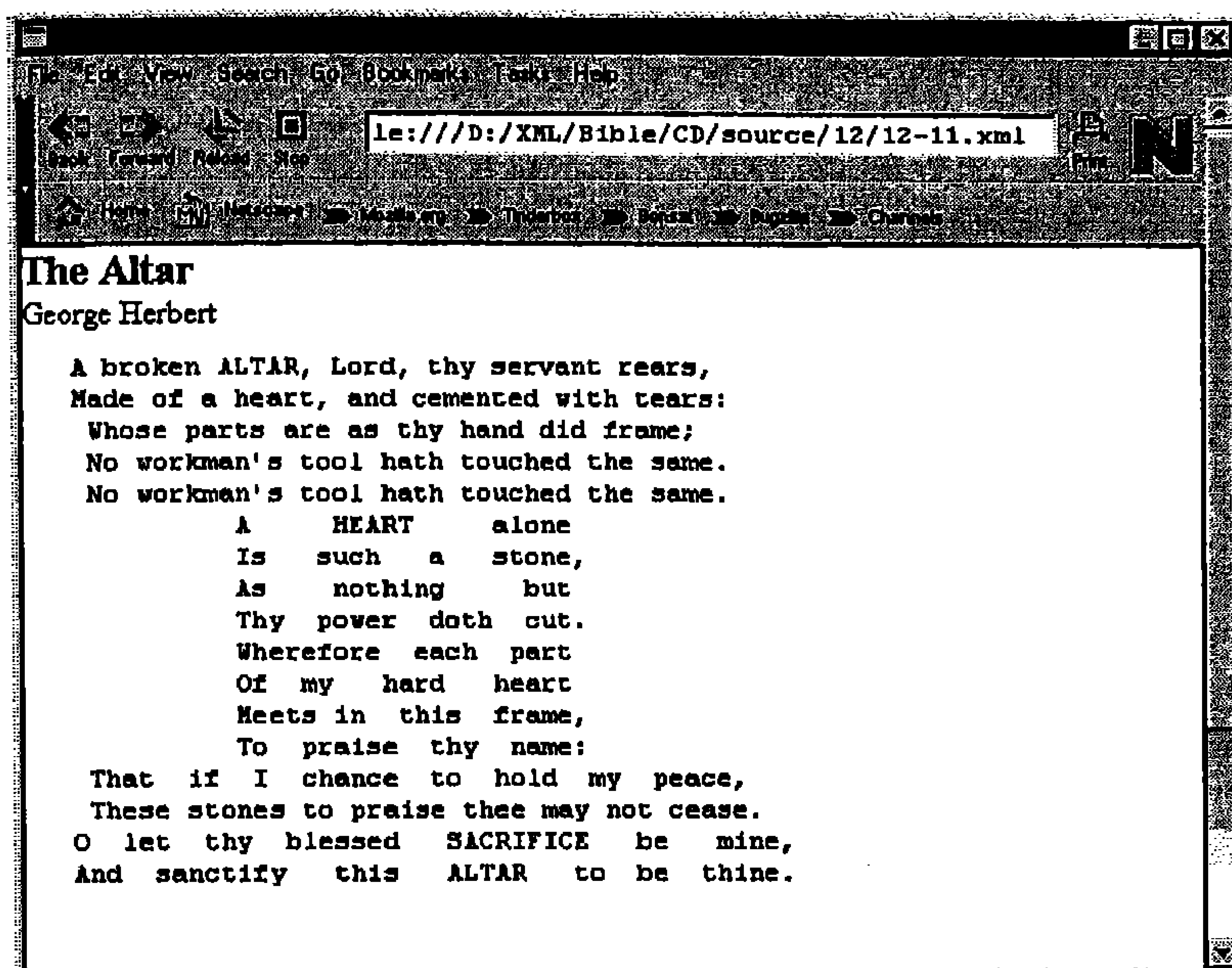
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="12-12.css"?>
<POEM>

<TITLE>The Altar</TITLE>
<POET>George Herbert</POET>

<VERSE>  A broken ALTAR, Lord, thy servant rears,</VERSE>
<VERSE>  Made of a heart, and cemented with tears:</VERSE>
<VERSE>   Whose parts are as thy hand did frame;</VERSE>
<VERSE>   No workman's tool hath touched the same.</VERSE>
<VERSE>   No workman's tool hath touched the same.</VERSE>
<VERSE>          A   HEART   alone</VERSE>
<VERSE>          Is  such  a  stone,</VERSE>
<VERSE>          As   nothing   but</VERSE>
<VERSE>          Thy  power  doth  cut.</VERSE>
<VERSE>          Wherefore  each  part</VERSE>
<VERSE>          Of  my   hard   heart</VERSE>
<VERSE>          Meets in  this  frame,</VERSE>
<VERSE>          To  praise  thy  name:</VERSE>
<VERSE>   That  if  I  chance  to  hold  my  peace,</VERSE>
<VERSE>   These stones to praise thee may not cease.</VERSE>
<VERSE>  O  let  thy  blessed  SACRIFICE  be  mine,</VERSE>
<VERSE>  And  sanctify  this  ALTAR  to  be  thine.</VERSE>

</POEM>
```

القامة ١٢-١٢ هي ورقة رمز تستخدم white-space: pre للحفاظ علـــى هــذا التكويــن. يوضح شكل ١٢-٥ النتائج معروضة على Mozilla.



الشكل ١٢-٥ قصيدة The Altar للشـــاعر جـــورج هربـــرت وبـــها white-space: pre

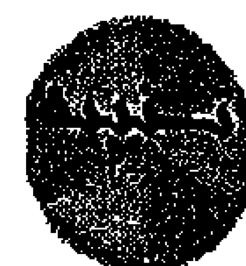


لا يدعم Explorer 5.0 خاصية white-space: pre

تعليمات البرمجة ١٢-١٢ ورقة الرمز للمسافات البيضاء الخاصة بالشعر

```
POEM   { display: block }
TITLE  { display: block; font-size: 16pt; font-weight: bold }
POET   { display: block; margin-bottom: 10px }
STANZA { display: block; margin-bottom: 10px }
VERSE  { display: block;
       White-space: pre; font-family: monospace }
```

وأخيراً قيمة عدم الالتفاف هي وسيلة تقوم بقسمة السطر في المكان الذي يوجد فيه الفاصل في النص الأصلي ولكنها تحدد مسافة واحدة في حالة عدم وجود فاصل. وقد تكون نافعة في حالة إذا ما كنت تريد إنتاج نص كلاسيكي يجب الحفاظ على أماكن الفواصل به أو إذا كنت تكتب قصائد شعرية تعتبر الفواصل عاملاً هاماً في تكوينها.



لا يدعم Explorer 5.0 وما يسبقه خاصية nowrap بأسلوب جيد.

خصائص حجم الخط

تدعم CSS Level 1 خمس خصائص أساسية لحجم الخط وهي:

1. font-family
2. font-style
3. font-variant
4. font-weight
5. font-size

وعلاوة على ذلك فإن خاصية font shorthand يمكنها إعداد الخصائص الخمسة في وقت واحد.

خاصية font shorthand

قيمة خاصية font shorthand هي قائمة من أسماء الخطوط تفصلها فواصل مثل Helvetica وtimes وPalatino إلى أخره. وتتضمن أسماء الخطوط المسافات البيضاء أيضاً مثل "Times New Roman" والتي يجب أن توضع بين قوسين.

وقد تكون الأسماء أيضاً أحد الأسماء الخمسة المنبثقة مثل: serif وsans-serif وcursive وfantasy وmonospace. ويستبدل المستعرض هذه الأسماء بنوع الخط المطلوب المثبت في النظام المحلي ويعرض الجدول ١٢-٤ هذه الأنواع من الخطوط.

الجدول ١٢-٤
أنواع الخطوط

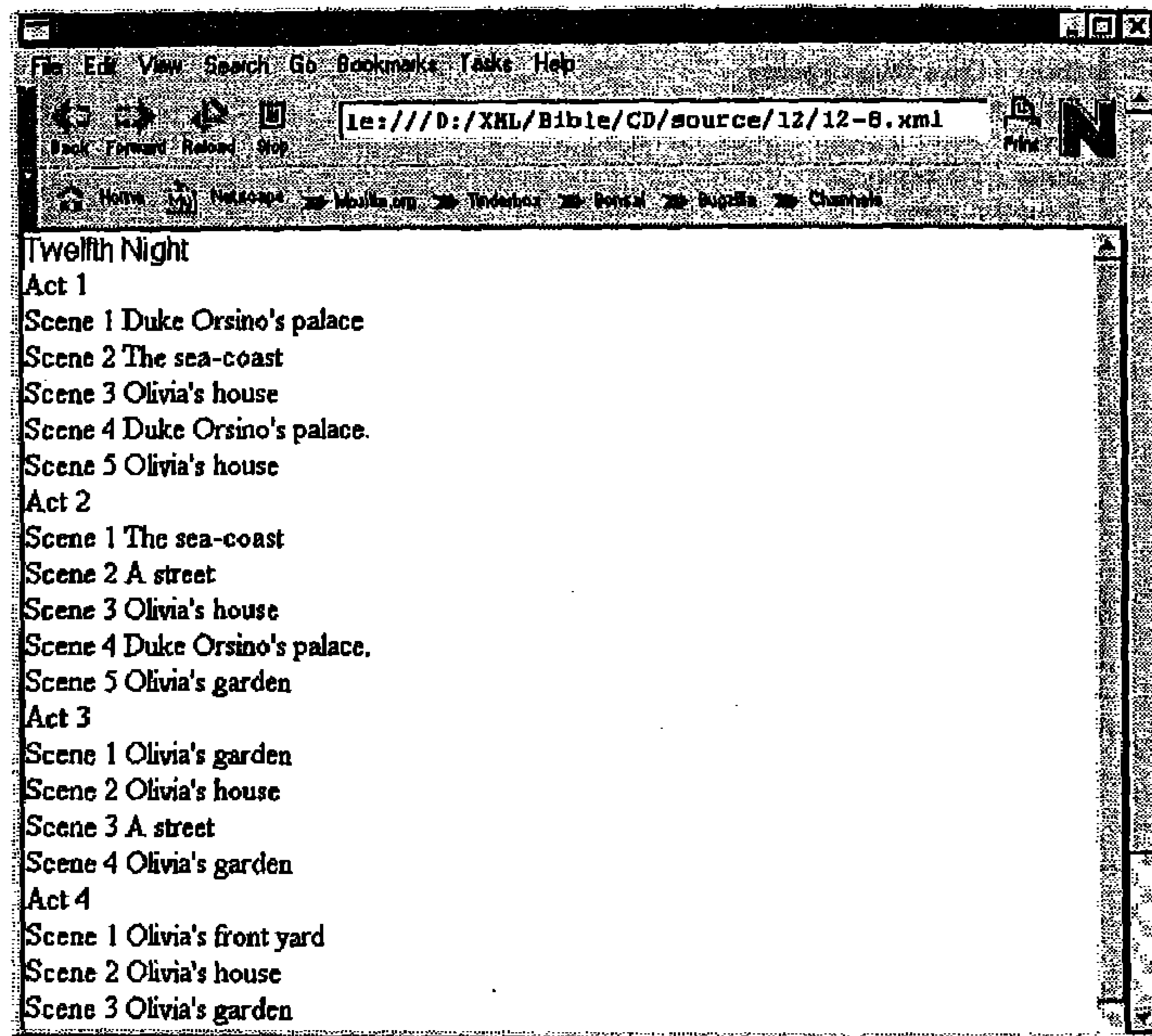
Example	Distinguishing Characteristic	Typical Family	Name
The quick brown fox jumped over the lazy dog.	Curlicues on the edges of letters make serif text easier to read in small body type.	Times, Times New Roman	Serif
The quick brown fox jumped over the lazy dog.	Block type, often used in headlines.	Geneva, Helvetica, Verdana	Sans-serif
The quick brown fox jumped over the lazy dog.	A typewriter like font in which each character has exactly the same width commonly used for source code and email.	Courier, Courier New, Monaco American Typewriter	Monospace
The quick brown fox jumped over the lazy dog.	Script font, a simulation of handwriting.	ZapfChancery	Cursive
The quick brown fox jumped over the lazy dog.	Text with special effects; e.g. letters on fire, letters formed by tumbling acrobats, letters made from animals, etc.	Western, Critter	Fantasy

ولأنه لا يوجد ضمان أن أي من أنواع الخطوط سوف يكون متوفراً أو مناسباً لنظام معين من العملاء "فمثلاً نظام ١٠ نقاط غير مسموح به في ماكينتوش وكذلك Palm Pilot" فأنك في الأغلب ستقوم بإعداد قائمة مفصولة بالفواصل بها الخيارات المتاحة للخط مرتبة حسب الأفضلية. وأخر خيار في القائمة يجب أن يكون أحد الأسماء أنواع الخطوط. ومع ذلك حتى إذا لم تحدد أسم نوع الخط ولم تكن الخطوط التي ترغب فيها موجودة فإن المستعرض سوف يقوم بالاختيار وقد يكون مختلف تمتماً عن رغبتك. فمثلاً فيما يلي قاعدتين لعنصر TITLE الأول من نوع Helvetica مائل للخلف والعناصر الأخرى Times في وضع مائل للخلف للنوع Times New Roman أو أي خط أخر من نوع serif.

```
TITLE    { font-family: Helvetica, sans-serif }
SYNOPSIS { font-family: Times, "Times New Roman", serif }
```

ويوضح شكل ١٢-٦ الملخص المحمل على Mozilla 5.0 بعد إضافة هاتين القاعدتين لورقة الرمز الموجودة في القائمة ١٢-٨. وليس هناك الكثير من التغييرات منذ القائمة ١٢-٤ ونوع الخط Times هو الخط الافتراضي. وأكثر التغييرات وضوحاً هو أن العنوان الآن مكتوب بـ Helvetica.

وخاصية font-family يتم توريثها للعناصر الفرعية وعليه فإنه عند إعداد SYNOPSIS باستخدام font-family للخط Times فإن كل العناصر التابعة سوف يتم إعدادها بنوع Times فيما عدا TITLE الذي تتجاوز خاصية font-family الخاصية التي تورثها.



الشكل ١٢-٦ ملخص مسرحية الليلة الثانية عشر مع العنوان بخط Helvetica.

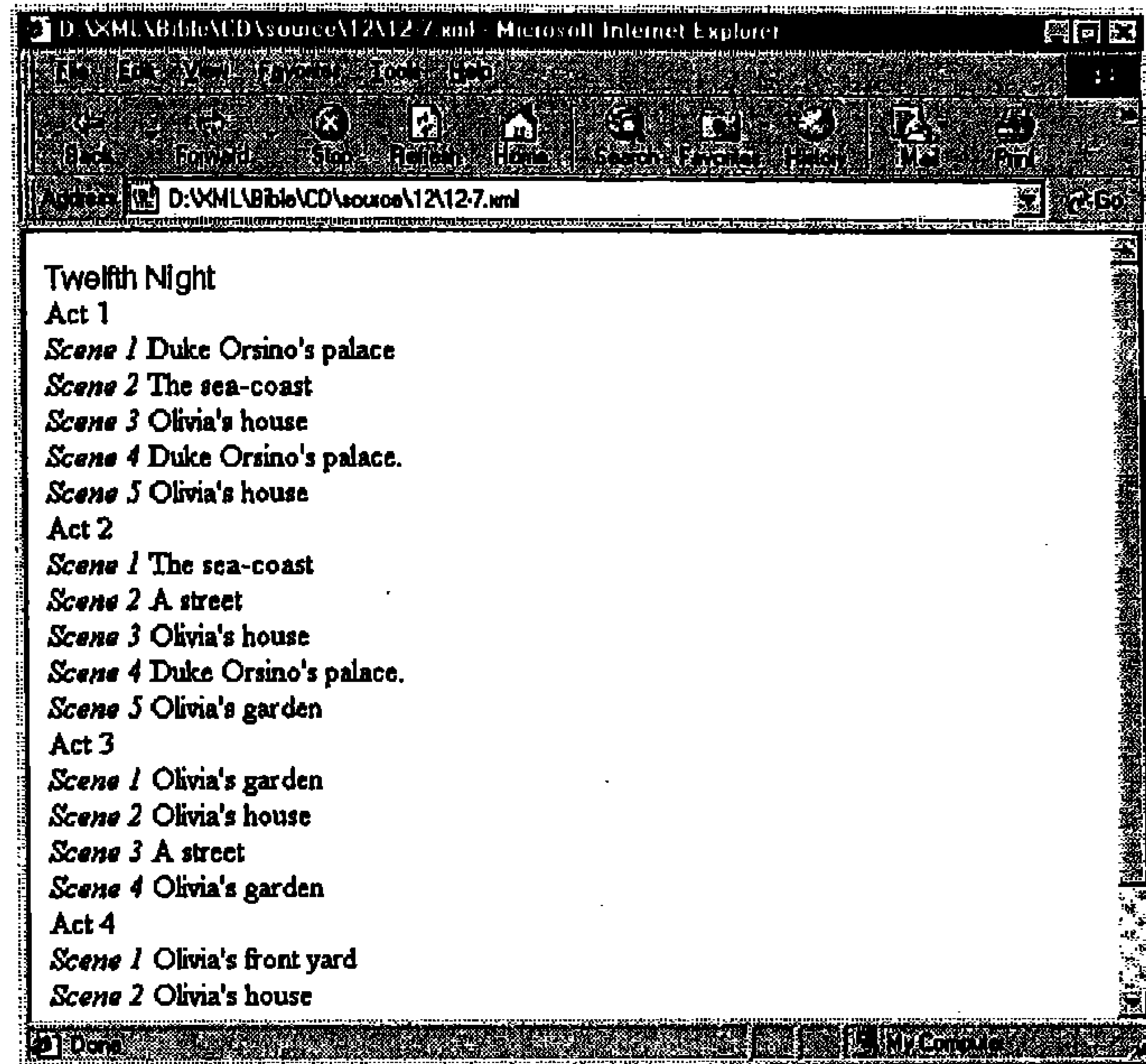
خاصية font-style

توجد ثلاث قيم لخاصية font-style وهي: normal وitalic وoblique والنص العادي الذي تقوم بقراءته الآن مكتوب بالخط normal. والناتج التقليدي لعنصر HTML EM هو النص ذو الخط المائل. أما النص Oblique فهو مماثل تماماً للنص ذو الخط المائل. ومع ذلك فإن نصوص Oblique يتم إنتاجها في النصوص التي تتبع النص simple algorithm للنص المائل العادي بكمية ثابتة. والنص ذو الخط المائل يستخدم نصاً تم تصميمه بحيث يبدو جيداً في الشكل المائل.

والقاعدة التالية تجعل خاصية SCENE_NUMBER مائلة:

```
SCENE_NUMBER { font-style: italic}
```

ويوضح الشكل ١٢-٧ الملخص الذي تم تحميله على Internet Explorer 5.0 بعد إضافة هذه القاعدة لورقة الرمز الخاصة به.



الشكل ١٢-٧ ملخص الليلة الثانية عشر باستخدام أرقام المشاهد مائلة

خاصية font-variant

توقد قيمتين ممكنتين لخاصية font-variant في CSS Level 1 وهما normal و small-caps . والوضع الافتراضي هو normal. وعند إعداد حجم الخط على small-caps يتم استبدال الحروف الصغيرة بحروف كبيرة مكتوبة بحجم صغير في النص الأصلي.

ويمكنك الحصول على نتيجة جيدة عن طريق دمج عنصر البيسودو first-letter مثلاً: عين عنصر ACT_NUMBER الذي سيحصل على font-variant: small-caps . وبعدها عين ACT_NUMBER الذي سيحصل على font-variant: normal وسوف تبدو الأرقام الناتجة للفصول كما يلي:

Act 1

وفيما يلي القاعدة:

```
ACT_NUMBER              { font-variant: small-caps}
ACT_NUMBER:first-letter { font-variant: normal}
```

وتتجاوز القاعدة الأولى القاعدة الثانية للرقم.

خاصية font-weight

تحدد خاصية font-weight نوع الخط من حيث كونه داكناً (bold) أو فاتحاً (narrow) ويظهر النص بثلاث عشرة قيم مختلفة لهذه الخاصية:

normal
bold
bolder
lighter
100
200
300
400
500
600
700
800
900

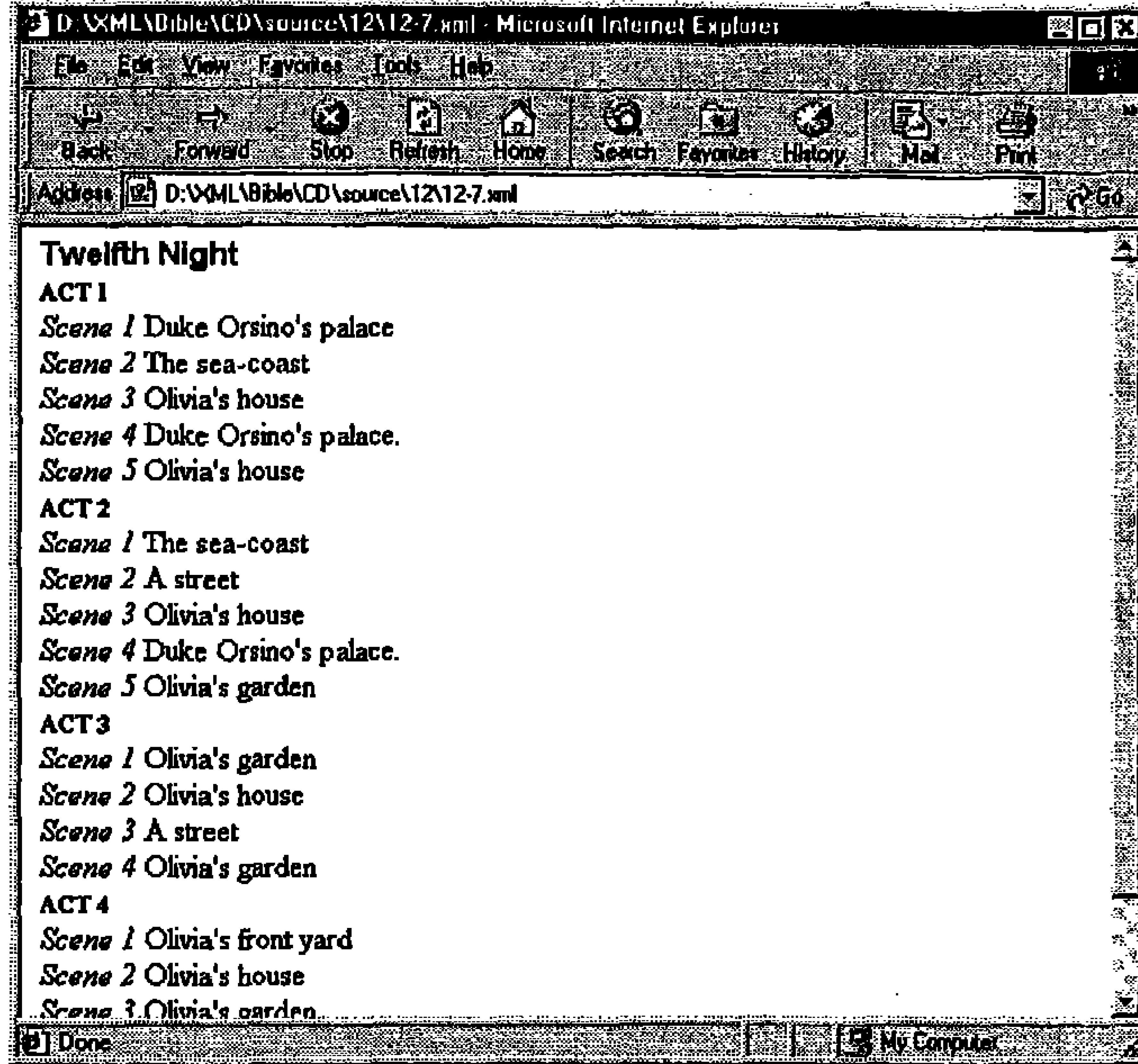
وتتراوح درجة ثقل الخط من ١٠٠ "وهي أفتحها" إلى ٩٠٠ "وهي أدكنها" ولا يسمح بالقيم المتوسطة مثل ٨٥٠. والدرجة التقليدية هي ٤٠٠ للفاتح و٧٠٠ للداكن. والعنصر bolder تجعل الخط أدكن من العنصر الأساسي. أما القيمة lighter تجعله يبدو أفتح من العنصر الأساسي. ومع ذلك فلا يوجد ضمان أن نوع خط معين له عدد تسعة درجات مختلفة من ثقل الخط. وفيما يلي قاعدة بسيطة تجعل عناصر TITLE وACT_NUMBER داكنة.

```
TITLE, ACT_NUMBER { font-weight: bold}
```

يوضح شكل ١٢-٨ النتائج على Mozilla بعد إضافة هذه القاعدة لورقة الرمز الموجودة في القائمة ١٢-٧.

خاصية font-size

تحدد خاصية font-size ارتفاع وعرض الحروف التقليدية في الخط والأحجام الكبيرة تأخذ مساحة أكبر على الشاشة. ويمكن أن يعتبر حجم الخط ككلمة أساسية أو كقيمة نسبية للعنصر الأساسي أو كنسبة مئوية لحجم العنصر الأساسي أو كرقم مطلق.



الشكل ١٢-٨ ملخص الليلة الثانية عشر باستخدام عنوان وأرقام المشاهد داكنة.

الكلمة الأساسية

الكلمات الأساسية للحجم هي:

xx-small
x-small
small
medium
large
x-large
xx-large

وهذه الكلمات الأساسية هي الأسلوب المفضل لإعداد أحجام الخط لأنها لا تزال متعلقة بحجم الخط الأساسي في الصفحة فمثلاً إذا قام المستخدم بإعداد حجم الخط الافتراضي على عشرون نقطة لأنه قصير النظر جداً فإن كل القيم الأخرى سوف تتغير بالتالي.

وكل حجم في CSS1 يضاعف أصغر حجم تالي بمقدار ١,٥مرة. والحجم الافتراضي هو medium وعليه فإنه إذا كان خط المستعرض الافتراضي ١٢ نقطة فإن نوع large سيكون ١٨ ويكون x-large جداً ٢٧ نقطة بينما الحجم xx-large سيكون 5.33 نقطة وحجم xx-small والذي بالكاد يكون مقبولاً 3.56 نقطة.

وفيما يلي القاعدة التي تجعل TITLE كبير جداً:

```
TITLE   { font-size: x-large }
```

القيم النسبية لحجم الخط الأساسي

يمكنك أيضاً تحديد الحجم النسبي مقارنةً بالعنصر الأساسي سواءً بحجم أصغر أو أكبر فمثلاً فيما يلي سوف يكون للعنصر SCENE_NUMBER حجم خط أصغر من حجم عنصره الأساسي SCENE.

```
SCENE_NUMBER { font-size: smaller }
```

وليس هناك قاعدة قاطعة عن المقدار الذي سيكون عليه العنصر أصغر أو أكبر بالتحديد وعموماً فإن المستعرض يحاول أن يتحرك من المتوسط إلى الصغير ومن الصغير إلى الصغير جداً وهكذا. والأمر بالمثل بالنسبة للأحجام الكبيرة. وعليه فحتى يصبح الخط أكبر فإنه يزداد بحوالي ٥٠ في المائة وجعله أصغر يقلل الحجم بحوالي٣٣ في المائة ولكن يترك للمستعرض تغيير هذه النسب حتى تتماشى مع الأحجام المختلفة.

نسبة حجم الخط بالنسبة إلى العنصر الأساسي

إذا لم تكن هذه الخيارات محددة بالقدر الكافي فبإمكانك استخدام تعديلات أكثر دقة عن طريق حجم الخط للعنصر الأساسي فمثلاً القاعدة التالية تعبر عن أن الحجم المستخدم للعنصر SCENE_NUMBER هو ٥٠ % من الحجم المستخدم للعنصر SCENE.

```
SCENE_NUMBER { font-size: 50% }
```

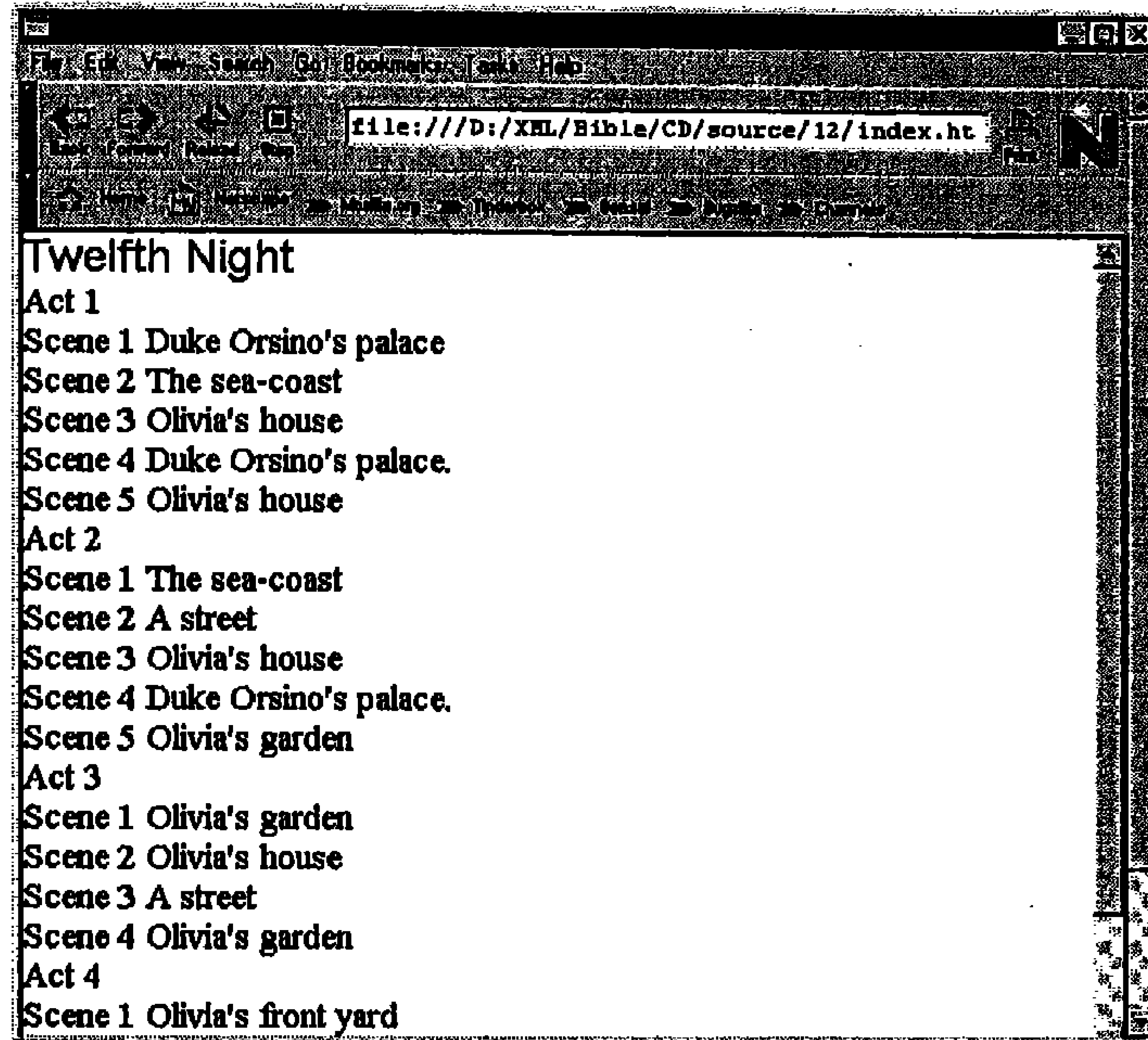
القيمة المطلقة للطول

وأخيراً يمكنك أن تعطي حجم الخط طولاً مطلقاً. وبالرغم من استطاعتك أن تستخدم البكسل والسنتيمترات أو البوصات فإن الوحدة الأكثر استخداما لقياس أحجام الخط هي النقط. فمثلاً تحدد القاعدة التالية font-size الافتراضي لعنصر SYNOPSIS وتوابعه بعدد ١٤ نقطة.

```
SYNOPSIS { font-size: 14pt }
```

من الأفضل عدم استخدام الوحدات المطلقة لحجم الخط لأنه من الصعب جداً "إن لم يكن من المستحيل" أن تنتقي طول معين يكون ملائماً للعرض على القواعد المختلفة التي ستعرض عليها الصفحات والتي تتراوح ما بين PDAs إلى Sony Jumbotron in Times Square وحتى عند تحديدها بأجهزة الكمبيوتر القياسية فإن معظم المصممين عادة غالبا ما يختاروا حجم خط صغير جداً في حين أن الخط يجب أن يكون ١٢ نقطة أو أكثر حتى يكون مقروءاً على الشاشة.

يوضح الشكل ١٢-٩ النتائج مكتوبة على Mozilla بعد إضافة هذه القاعدة لورقة الرمز الموجودة في القائمة ١٢-٧. ونص المشاهد غير مكتوب بخط أكبر في واقع الأمر ولكنه في الحقيقة أكبر فقط وأسهل في القراءة في جميع الأحوال.



الشكل ١٢-٩ ملخص مسرحية الليلة الثانية عشر باستخدام أحجام خط مختلفة.

خاصية font Shorthand

خاصية font Shorthand هي خاصية تتيح أن يتم إعداد كل من نمط الخط وثقله وحجمه وكذلك عائلته في قاعدة واحدة وفيما يلي قاعدتين لكل من عنصري TITLE وSCENE_NUMBER والتي كانت متفرقة في ستة قواعد مختلفة من الجزء السابق.

```
TITLE { font: bold x-large Helvetica, sans-serif }
SCENE_NUMBER { font: italic smaller serif }
```

ويجب أن تعطى القيم بالترتيب التالي:

١- كل من النمط والتغير والثقل بأي ترتيب وأياً منها يمكن أن تتم إزالته.

٢- الحجم الذي لا يتغير.

٣- توضع اختياريا شرطة مائلة وارتفاع للسطر.

٤- أسم العائلة والذي لا يمكن إزالته.

إذا كان هذا الأمر يبدو معقداً فهذه هي الحقيقة فمن الصعب تذكر الترتيب الصحيح لهذه الخصائص دون الرجوع إليها لذلك فمن الأفضل استخدام الخصائص كل على حدة.حيث أنه من غير المؤكد أن أسلوب الاختصار هذا سوف يوفر الوقت.

وتعبر القائمة ١٢-١٣ ورقة الرمز لملخص المسرحية التي تم إجراء القواعد عليها حتى الآن وذلك باستخدام خواص اختصار font وهي تساوي تماماً مجموع القواعد التي تتكون منها منفصلة. وليس هناك أي تغيير في المستند الناتج.

تعليمات البرمجة ١٢-١٣: ورقة الرمز لملخص المسرحية باستخدام اختصار font

```
SYNOPSIS, TITLE, ACT, SCENE { display: block }
SCENENUMBER { font: italic smaller serif }
TITLE { font: bold x-large Helvetica, sans-serif }
SYNOPSIS { font: 14pt Times, زTimes New Romanس, serif }
ACTNUMBER { font-variant: small-caps}
ACTNUMBER:first-letter { font-variant: normal}
ACTNUMBER { font-weight: bold}
```

خاصية اللون

تسمح لك CSS بتحديد اللون لأي العناصر التي ترغب فيها على الصفحة. وتتكون قيمة خاصية color من ١٦ لون التي تم ذكر كلماتها الأساسية أو من الكسور العشرية للثلاثية RGB أو الأجزاء السدس عشرية وعلى سبيل المثال توضح القاعدة التالية أن كل العناصر قد تلونت باللون الأسود فيما عدا SCENE_NUMBER الذي تلون باللون الأزرق:

```
SYNOPSIS { color: black }
SCENE_NUMBER { color: blue}
```

ويتم توريث خاصية اللون للعناصر التابعة وبالتالي فإن كل العناصر في الملخص فيما عـــدا SCENE_NUMBER ستكون باللون الأسود.

والقواعد التالية كلها معادلة للقاعدتين العلويتين ومن الأفضل استخدام أسماء الألـــوان كلمـــا أمكن ذلك. وإذا لم يتاح ذلك فيفضل استخدام مستعرض خاص بالألوان.

```
SYNOPSIS { color: #000000 }
SCENE_NUMBER { color: #0000FF}
SYNOPSIS { color: rgb(0, 0, 0) }
SCENE_NUMBER { color: rgb(0, 0, 255)}
SYNOPSIS { color: rgb(0%, 0%, 0%) }
SCENE_NUMBER { color: rgb(0%, 0%, 100%)}
```

خصائص الخلفية

يمكن إعداد خلفية العنصر لعرض لون أو صورة. وفي حالة إعدادها لعض الصورة فيمكن وضع الصورة بأوضاع مختلفة بما يتناسب مع محتوى العنصر ويتحقق ذلك بالقواعد الخمسة التالية:

١- background-color

٢- background-image

٣- background-repeat

٤- background-attachment

٥- background-position

وأخيراً توجد خاصية اختصار لـــ background والتي تسمح لك بإعداد بعض أو كل هـــذه الخصائص باستخدام قاعدة واحدة.



يتم استخدام الخلفيات المبهرة بكثرة على الويب هذه الأيام وعند اســـتخدامك لأي لون غير ساطع في الخلفية سوف يجعل صفحتـــك صعبـــة القـــراءة ومزعجـــة للمستخدم. لذلك فإن إدراج هذه العناصر هنا كان لغرض الشمولية فقط ولكن مـــن الأفضل الاقتصاد في استخدامها.

لا يتم توريث خصائص الخلفية. وعلي كل عنصر فرعي أن يحدد صفاته الخاصة بها. ومع ذلك فقد تبدو خصائص الخلفية وكأنها موروثة نتيجة لأن الخلفية الافتراضية هي الخلفية الشفافة ومن ثم فإن أي خلفية أي خلفية موجودة تحت العنصر تظهر من خلالها وفي معظم الأحوال تكون مجرد خلفية العنصر الأساسي.

خاصية background-color

يمكن إعداد خاصية background-color لتكون بنفس القيم مثل خاصية color. ولكن بدلاً من استبدال لون محتوى العنصر فهي تغير لون خلفية العنصر التي سيعرض عليها فمثلاً لكي ترسم عنصر SIGN ذو نص أصفر على خلفية زرقاء يمكن استخدام القاعدة التالية:

```
SIGN { color: yellow; background-color: blue}
```

ويمكنك أيضاً إعداد الخلفية على الكلمة الأساسية transparent "الوضع الافتراضي" والتي تعني أن الخلفية ستكون بلون أو صورة العنصر التي توجد فوقه. وعادة ما يكون العنصر الأساسي.

خاصية background-image

تعد خاصية background-image على إما none "وهو الوضع الافتراضي" أو "URL" "وهو الوضع النسبي" حيث توجد صورة ذات نقط مضيئة في الملف. وإذا كانت URL فأن المستعرض يحمل الصورة ويقوم باستخدامها في الخلفية مثل خاصية BACKGROUND لعنصر BODY المستخدم في HTML. فمثلاً فيما يلي كيف ستتمكن من إضافة ملف party.gif "المعروض في الشكل ١٢-١٠" والخاص بخلفية عنصر INVITATION

```
INVITATION { background-image: url(party.gif) }
```



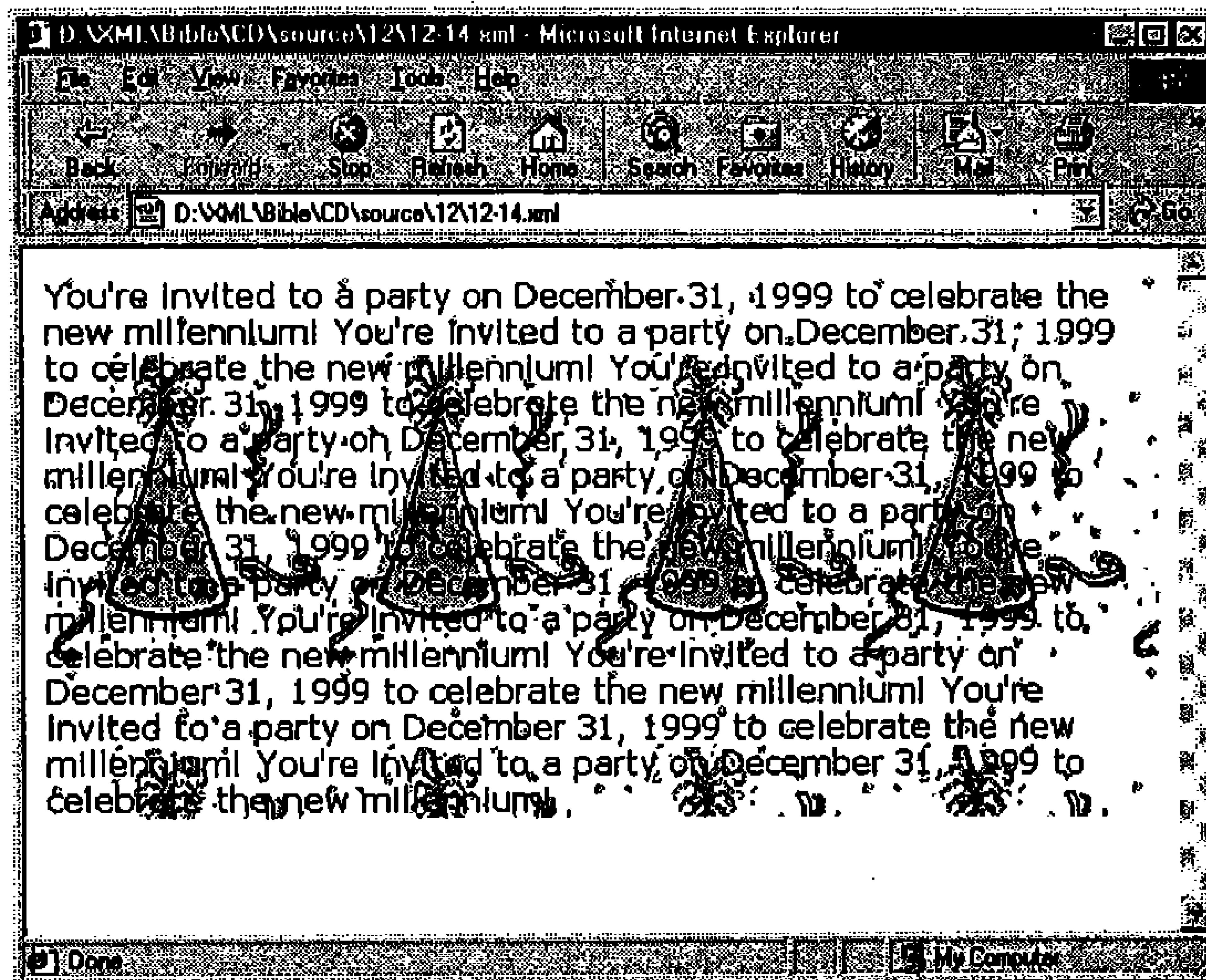
الشكل ١٢-١٠ الخلفية الأصلية بدون أي تغييرات أو تعديلات لدعوة الحفلة الموجودة في القائمة ١٢-١٤.

وتم رسم الصورة التي تسمىbackground-image تحت العنصر المحدد وليس تحت عارضة المستعرض مثل ميزة BACKGROUND الخاصة بالعنصر BODY في HTML.

ملحوظة

وإذا كانت الصورة في الخلفية مرتبطة بعنصر جذري فالإصدارات المبكرة من Mozilla 5.0 تربط صورة الخلفية بعارضة المستند كله بدلاً من العنصر وحده. ولكن بالنسبة لكل العناصر الغير جذرية تطبق صورة الخلفية على العنصر المخصصة له فقط. ولم تحدد مخصصات CSS Level 1 ما إذا كانت تقوم بهذا الأجراء أم لا.

وفي الأغلب لن تكون صور الخلفية بنفس حجم محتويات الصفحة. وإذا كانت الصورة أكبر من مربع العنصر فسوف تبدو الصورة مكومة أما إذا كانت أصغر من مربع العنصر فسوف تبدو بها مربعات رأسية وأفقية ويوضح الشكل ١٢-١١ صورة خلفية تغطي المحتويات التي تحتها ولاحظ أن التغطية تكون فوق العنصر وليس بملء الشاشة. ويظهر ملف XML لهذه الصورة في القائمة ١٢-١٤.



الشكل ١٢-١١ صورة خلفية تغطي النص.

تعليمات البرمجة ١٢-١٤: بطاقة دعوة لحفل على XML

```
<?xml version=1.0?>
<?xml-stylesheet type=text/css href=party.css?>
<INVITATION>
Your invited to a party on December 31, 1999 to celebrate the
```

new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium! You're invited to a party on December 31, 1999 to celebrate the new millennium!
</INVITATION>

خاصية background-repeat

تقوم خاصية background-repeat بتعديل المساحة التي تشغلها الصورة على الشاشة. ويمكنك تحديد أن تشغل الصورة مساحة رأسية أو أفقية والقيم والخصائص الممكنة هي:

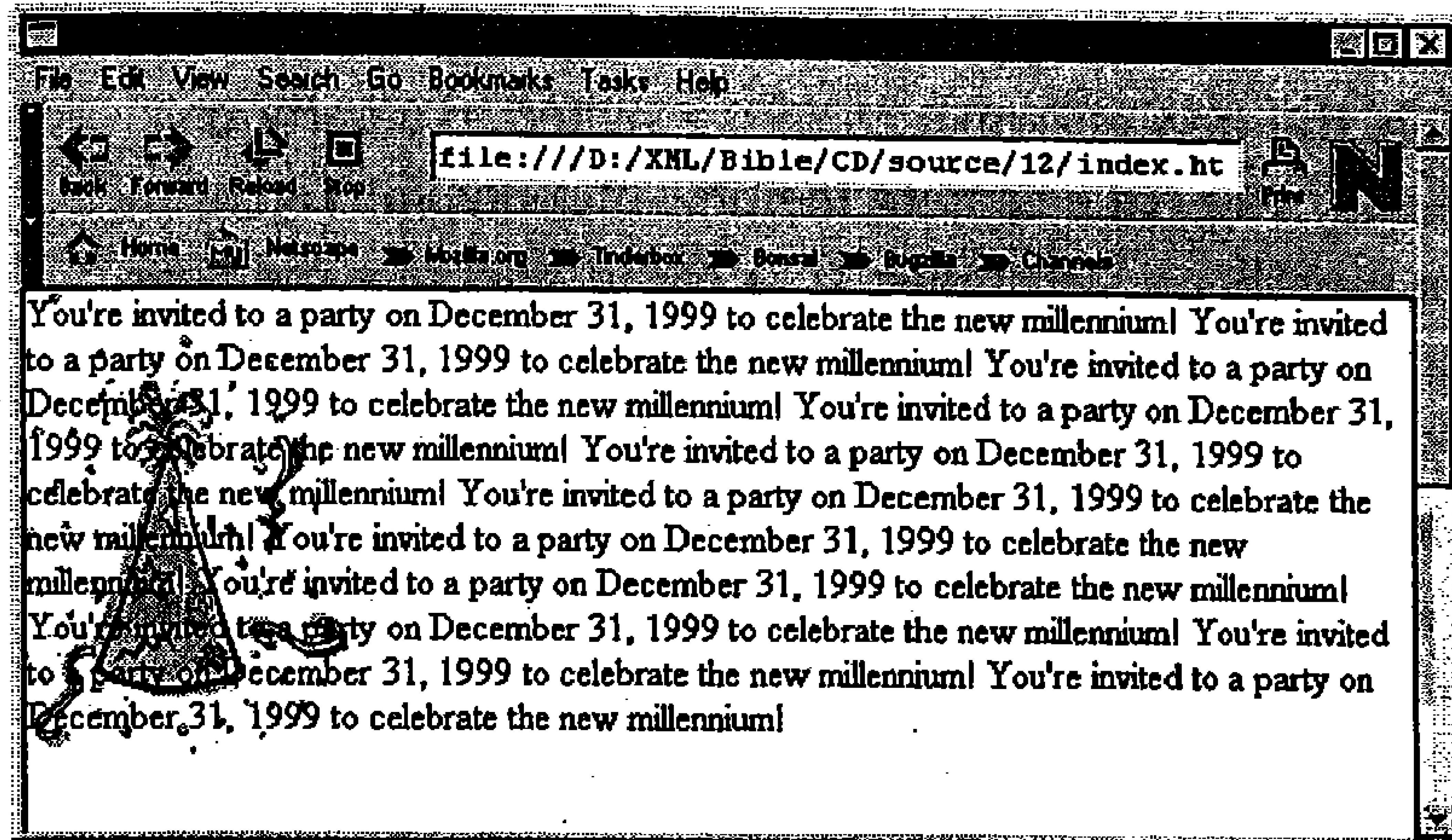
repeat
repeat-x
repeat-y
no-repeat

ومثلاً إذا أردت ظهور قبعة حفلات واحدة في بطاقة الدعوة فإنك سوف تقوم بإعداد background-repeat الخاص بعنصر INVITATION على no-repeat. ويوضح شكل ١٢-١٢ النتائج مثلاً:

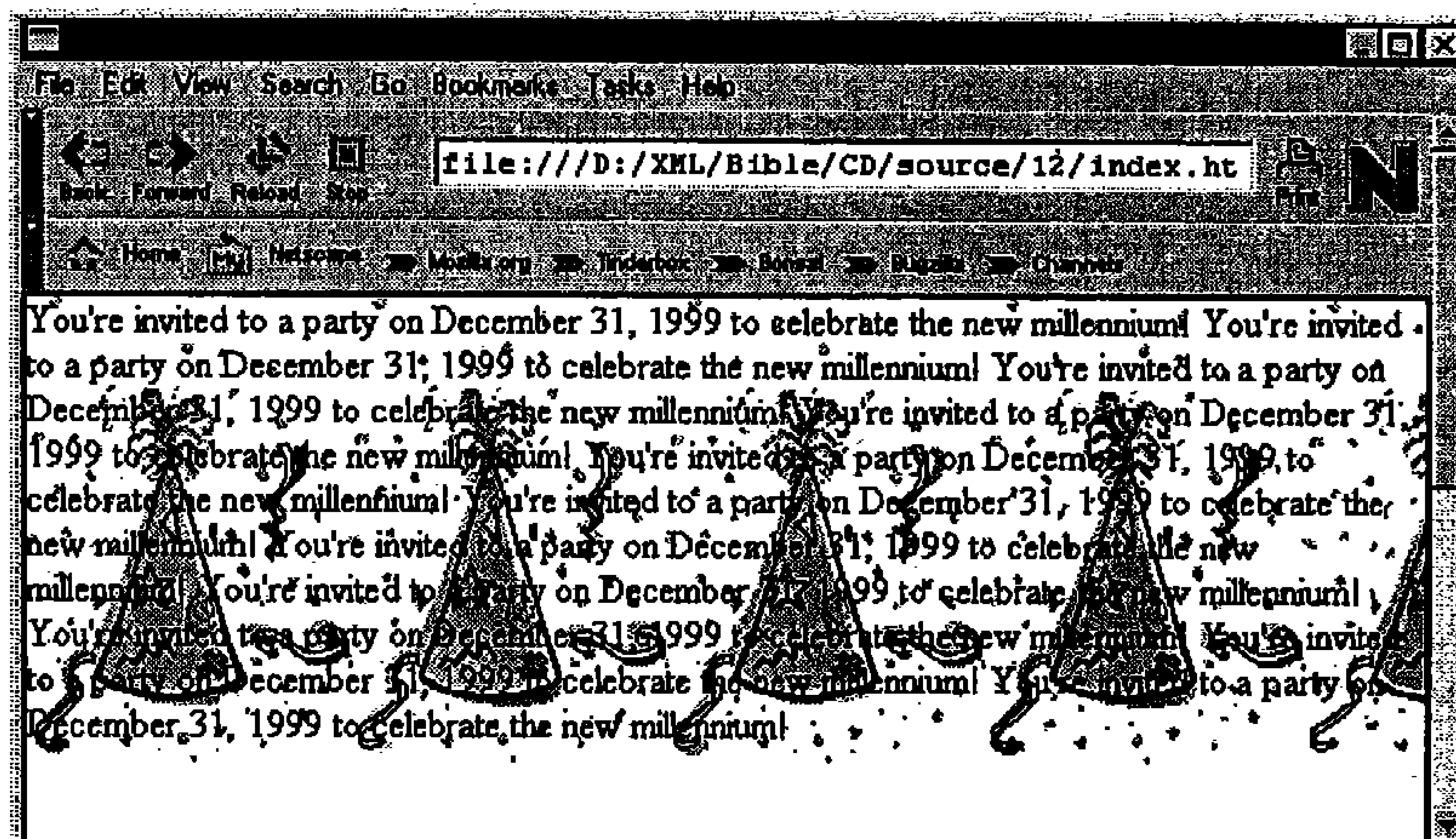
```
INVITATION { background-image:  url(party.gif);
             background-repeat: no-repeat }
```

لكي تشغل الصورة المساحة بعرض الصفحة وليس من أعلى إلى أسفل قم بإعداد background-repeat على repeat-x كما هو موضح فيما يلي. ويوضح شكل ١٢-١٣ صورة خلفية تشغل المساحة بالعرض وليس طولياً.

```
INVITATION { background-image:  url(party.gif);
             background-repeat: repeat-x }
```



الشكل ١٢-١٢ صورة خلفية لا تشغل المساحة كلها.



الشكل ١٢-١٣ صورة خلفية تتكرر بعرض الصفحة وليس بأسلوب طولي.

ولكي تجعل الصورة تشغل من أعلى إلي أسفل وليس بالعرض أعد background-repeat على repeat-y كما هو موضح فيما يلي ويعرض الشكل ١٢-١٤ النتائج.

```
INVITATION { background-image:  url(party.gif);
          background-repeat: repeat-y }
```

خاصية background-attachment

تلحق الصورة بالمستند في HTML وعند يستعرض المستند تعرض معه صورة الخلفية. وباستخدام خاصية background-attachment يمكنك تعيين إذا ما كانت الصورة مرتبطة

بإطار أو جزء أخر. والقيم الممكنة هي scroll وfixed والوضع الافتراضي هو scroll وهذا ما يعني أن الصورة مرتبطة بالمستند بدلاً من الإطار.



الشكل ١٢-١٤ صورة خلفية تعرض في اتجاه لأسفل بدلاً من العرض.

ولكن عند إعداد background-attachment على fixed يستعرض المستند بينما تبقى الصورة ثابتة وقد يكون ذلك مفيداً في حالة الصور الكبيرة التي يناسب حجمها حجم المستعرض والقاعدة التي تحتاجها كما يلي:

```
DOCUMENT { background-attachment: fixed;
         background-repeat: no-repeat }
```



لا تقوم IE5 ولا Mozilla بتدعيم الخلفية الثابتة وهذه الخاصية يمكن أن تضاف إلى الإصدارات التالية "لا تتطلب CSS1 spec إلى مستعرضات تدعم الخلفيات الثابتة".

خاصية background-position

افتراضياً يرتبط الركن العلوي الأيسر للعنصر مع الركن الأيسر العلوي للمستند. "وكمثال أنظر للشكل ١٢-١٢" وفي معظم الأحوال يكون هذا بالتحديد ما تحتاجه. ولكن في حالات النادرة التي ستحتاج فيها غير ذلك سوف تسمح لك خاصية background-position بتحريك الخلفية بالنسبة للعنصر.

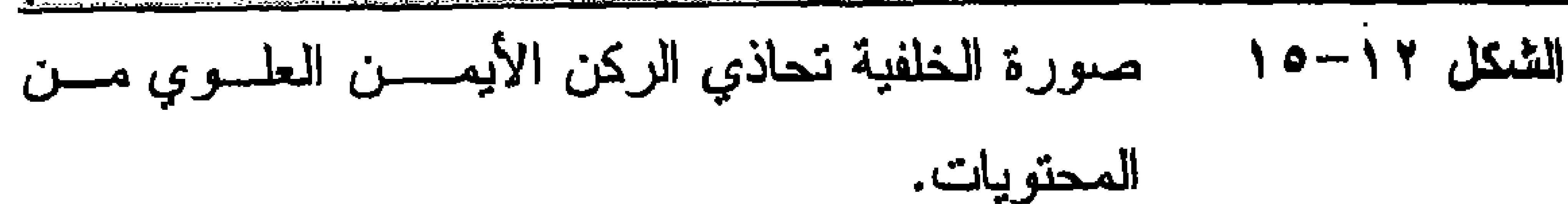
ويمكنك أن تحدد باستخدام النسبة المئوية لكل من العرض والطول للعنصر الأساسي وذلك باستخدام العرض المطلق أو باستخدام أي أثنين من الكلمات الأساسية التالية:

top
center
bottom
left
center
right

النسبة المئوية لعرض وارتفاع العنصر الأساسي

تسمح لك النسبة المئوية أن تثبت الأجزاء المختلفة من الخلفية على جزء العنصر الذي يناسبها ويتم إعطاء المعامل x على أنه النسبة المئوية التي تتراوح ما بين 0% من ناحية اليد اليسرى إلى 100% ناحبة اليد اليمنى. ويرمز المعامل y إلى النسبة المئوية التي تتراوح ما بين 0% من الجهة العلوية ونسبة 100% من أسفل. ومثلاً تضع القاعدة التالية الركن العلوي الأيمن من الصورة في الجزء العلوي الأيمن من عنصر INVITATION. ويوضح الشكل ١٢-١٥ هذه النتائج:

```
INVITATION { background-image: url(party.gif);
           background-repeat: no-repeat;
           background-position: 100% 0% }
```

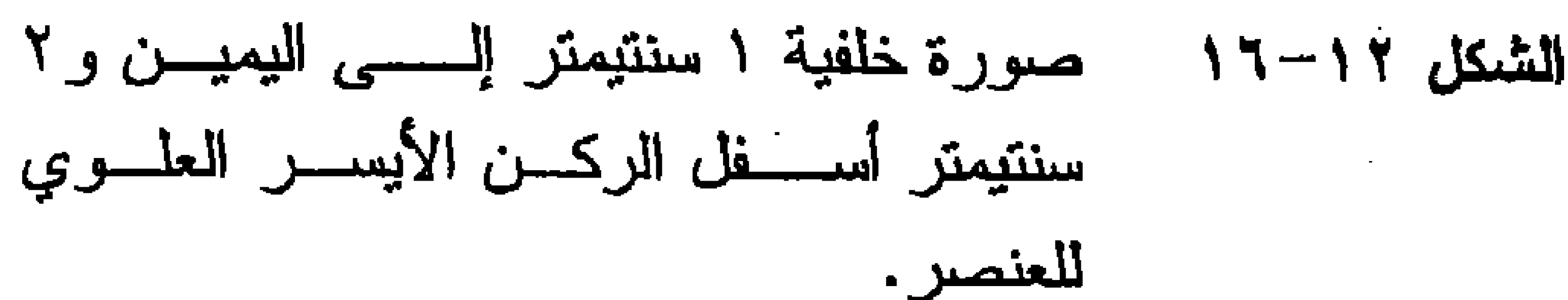


الشكل ١٢-١٥ صورة الخلفية تحاذي الركن الأيمن العلوي من المحتويات.

الطول المطلق

يضع الطول المطلق الركن العلوي الأيسر للخلفية في وضع مطلق بالنسبة للعنصر. والقاعدة التالية تضع الركن الأيسر العلوي لصورة الخلفية party.gif سنتيمتر واحد إلى اليمين وسنتيمترين أسفل الركن العلوي الأيسر للعنصر ويوضح الشكل ١٢-١٦ النتائج.

```
INVITATION { background-image: url(party.gif);
             background-repeat: no-repeat;
             background-position: 1cm 2cm }
```



الشكل ١٢-١٦ صورة خلفية ١ سنتيمتر إلى اليمين و ٢ سنتيمتر أسفل الركن الأيسر العلوي للعنصر.

الكلمات الأساسية

تتماثل الكلمات الأساسية لكل من top left وtop left وهما 0% 0%. وكل من top وtop center وcenter top وهي 50% 0%. وكل من right top وtop right هما 0% 100% وكلمات The left وleft center وcenter left هي 0% 50% وكلمات center وcenter وcenter 50% 50% بينما كلمات right وright center وcenter right هي 50% 100% وكلمات bottom left وleft bottom هي 100%، 0% بينما تعني كل من bottom وbottom وcenter bottom وcenter نفس الشيء وهو 50% 100%. أما بالنسبة للكلمات bottom right وright bottom فهي تحمل ذات المعني وهو 100% 100%. ويربط الشكل ١٢-١٧ هذه الكلمات بالأوضاع المنفردة لمربع العنصر.

top left left top 0% 0%	top top center center top 50% 0%	top right right top 100% 0%
left center left left center 0% 50%	center center center 50% 50%	right center right right center 100% 50%
bottom left left bottom 0% 100%	bottom bottom center center bottom 50% 100%	bottom right right bottom 100% 100%

الشكل ١٢-١٧ الأوضاع النسبية لصورة الخلفية.

ومثلاً في مثالنا الحالي لدعوة الحفلة أفضل النتائج تكون بتثبيت المركزين معاً كما هو موضح بالشكل ١٢-١٨ وفيما يلي القاعدة الخاصة بذلك:

```
INVITATION { background-image: url(party.gif);
             background-repeat: no-repeat;
             background-position: center center }
```



الشكل ١٢-١٨ صورة لا تملئ كل الخلفيـــة مثبتـــة فـــي منتصف عنصر INVITATION

وإذا كانت قيمة الخاصية background-attachment هي fixed فإنه سيتم وضع الصــورة في إطار الصندوق بدلاً من العنصر.

خاصية Background Shorthand

هذه الخاصية هي اختصار لإعداد خصـــائص background-color وbackground-image وbackground-image وbackground-attachment وbackground-position في قاعدة واحدة. فمثلاً لكي تقوم بإعداد background-color على اللون الأبيض وbackground-color على party.gif وbackground-repeat علـــى no-repeat وbackground-attachment على fixed في عنصر INVITATION يمكنك استخدام القاعدة التالية:

```
INVITATION { background: url(party.gif) white no-repeat fixed }
```

وهذا يعني نفس الشيء للقاعدة التالية التي هي أطول ولكنها أكثر قابلية للتنفيذ:

```
INVITATION { background-image: url(party.gif);
          background-color: white;
          background-repeat: no-repeat;
          background-attachment: fixed }
```

وعند استخدام خاصية background shorthand يمكن إعطاء القيم الخاصة بأي واجدة من الخصائص الخمسة بأي ترتيب ولكن بحيث لا تتكرر أياً منها أكثر من مرة واحدة. فمثلاً محـــاذاة الركن العلوي الأيمن المستخدم في الشكل ١٢-١٦ تكتب بهذا الأسلوب:

```
INVITATION { background: url(party.gif) no-repeat 100% 0% }
```

خصائص النص

هناك ثمانية خصائص تؤثر على مظهر النص بخلاف الخط وهي:

١- word-spacing

٢- letter-spacing

٣- text-decoration

٤- vertical-align

٥- text-transform

٦- text-align

٧- text-indent

٨- line-height

خاصية word-spacing

تمد خاصية word-spacing النص بإضافة مسافات إضافية بين الكلمات. وتزيل القيمة الســـالبة المسافات بين الكلمات. والحالة الوحيدة التي ربما تحتاج عندها تغيير المسافات البينية للويب هـــي حالة الطالب الذي يعمل من خلال عدد صفحات محدود ويود أن يبدو عمله أكـــبر أصغـــر مـــن الحقيقة.

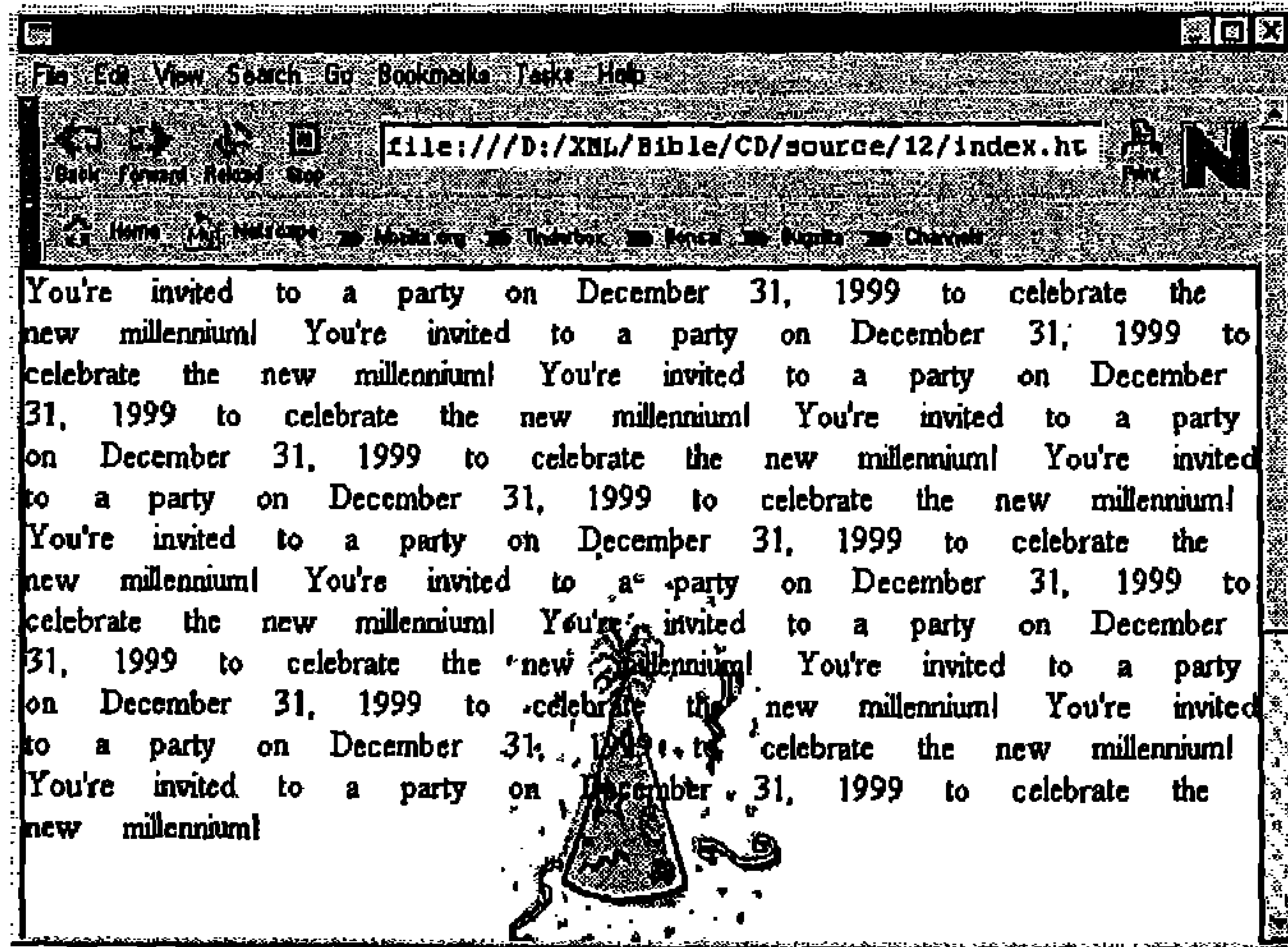
يحب ناشري الشاشات الرئيسية تغيير هذه الخصائص والمشـــكلة أن كــل هــذه القواعد التي تعلمتها عن كيفية ومتى تعدل المسافات لا تعدو أن تكون حبراً علـــى ورق ولا تعمل في الواقع عند الانتقال إلى الإلكترونات والمورفونات المتوســـطة "الشاشة CRT التقليدية" والحل الأفضل هو أن تترك للمستعرض اتخاذ القـــرارات بشأن المسافات البينية.

والغرض الأساسي من رؤية النتائج على الورق بالفعل هو التأكد مـــن أن حجـم الخط بالفعل وارتفاع الكلمات وعدد النقاط التي تم استخدامها في البوصة ة ما إلى ذلك. وفي حالة الويب يكون لديك قدر كافي من المعلومات عن الوســـط النـــاتج والمتاح للتحكم في كل هذه التفاصيل.

ولكي تغيير الوضع الافتراضي للقيمة من normal فإنك تحدد طول للخاصية مثل:

```
INVITATION { word-spacing: 1em }
```

ولا يطلب من المستعرضات أن تحترم هذه الخاصية وبالأخص إذا تعارضت مع خصـــائص أخرى مثل، align: justified ولا يدعم Internet Explorer 5.0 خاصيـــة word-spacing ولكن تدعمها Mozilla كما هو موضح في الشكل ١٢-١٩.



الشكل ١٢-١٩ عنصـــر INVITATION مـــع 1em مـــن المسافات البينية.

خاصية letter-spacing

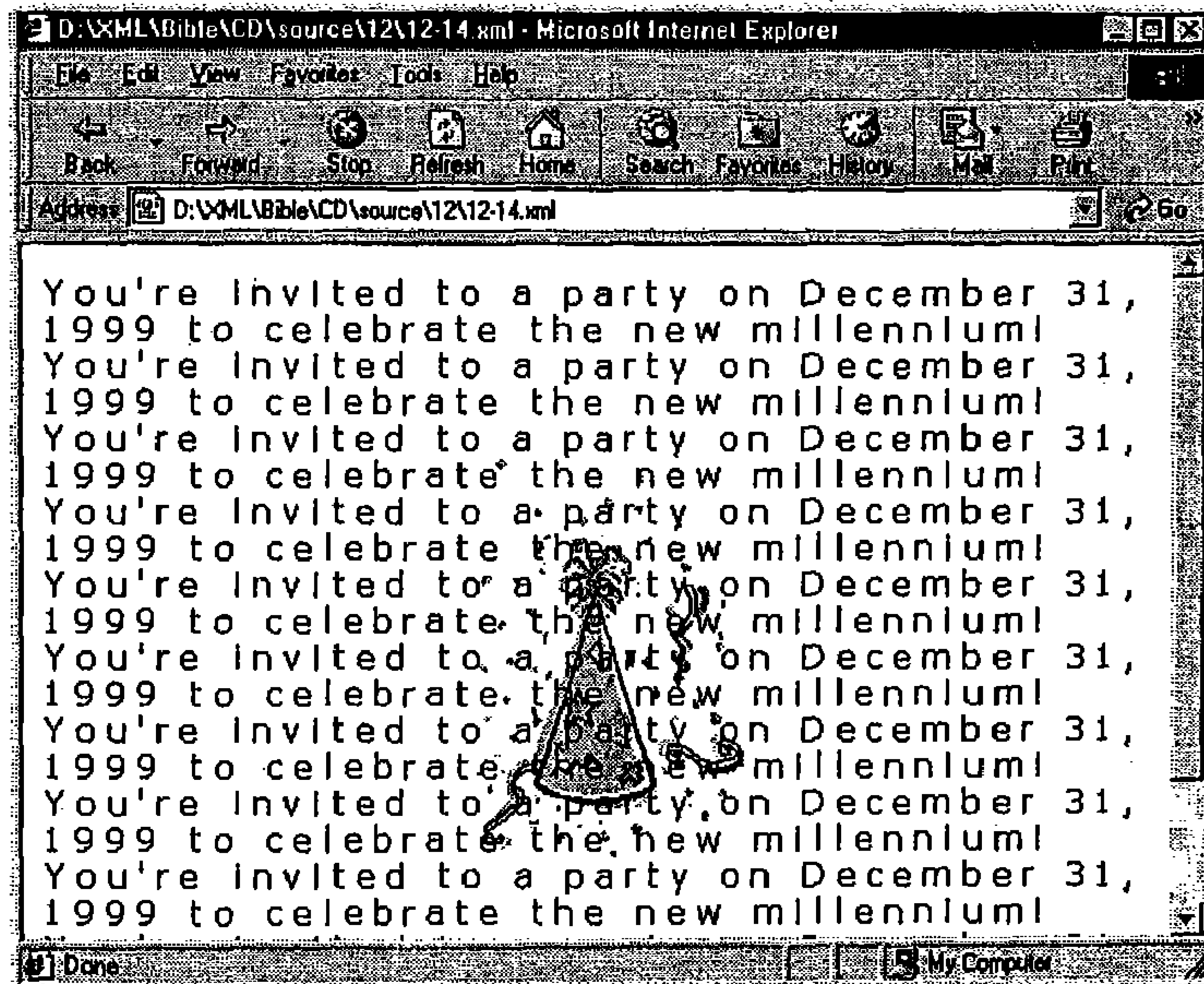
تسمح لك خاصية letter-spacing بأن يتمدد النص وذلك بإضافة مسافات إضافية بين الحروف. وعند إعداد القيمة السلبية تقوم بإزالة المسافات بين الحروف ومرة أخرى فإن السبب الوحيد لاستخدام هذه الخاصية كما أسلفت أن يكون المستخدم طالباً ويحاجة لإخراج عمله في نطاق عدد محدود من الصفحات.

لكي تقوم بتغيير الوضع الافتراضي إلى القيمة normal عليك تحديد الطول للخاصية مثل:

```
INVITATION { letter-spacing: 0.3em }
```

وحيث أن المحاذاة تعمل بتعديل المسافات بين الحروف فإن تغيير المسافات يدوياً يمنع المستعرض من محاذاة النص.

ولا يتحتم على المستعرضات احترام هذه الخاصية وبخاصة إذا تعارضت مع خصائص أخرى مثل: align: justified ولكن تدعمها كل من Internet Explorer وMozilla كما هو موضح في الشكل ١٢-٢٠.



الشكل ١٢-٢٠ عنصر INVITATION بوجود مسافة 0.3em بين الحروف.

خاصية text-decoration

تكون لخاصية text-decoration أحد القيم الخمسة التالية:

none
underline
overline
line-through
blink

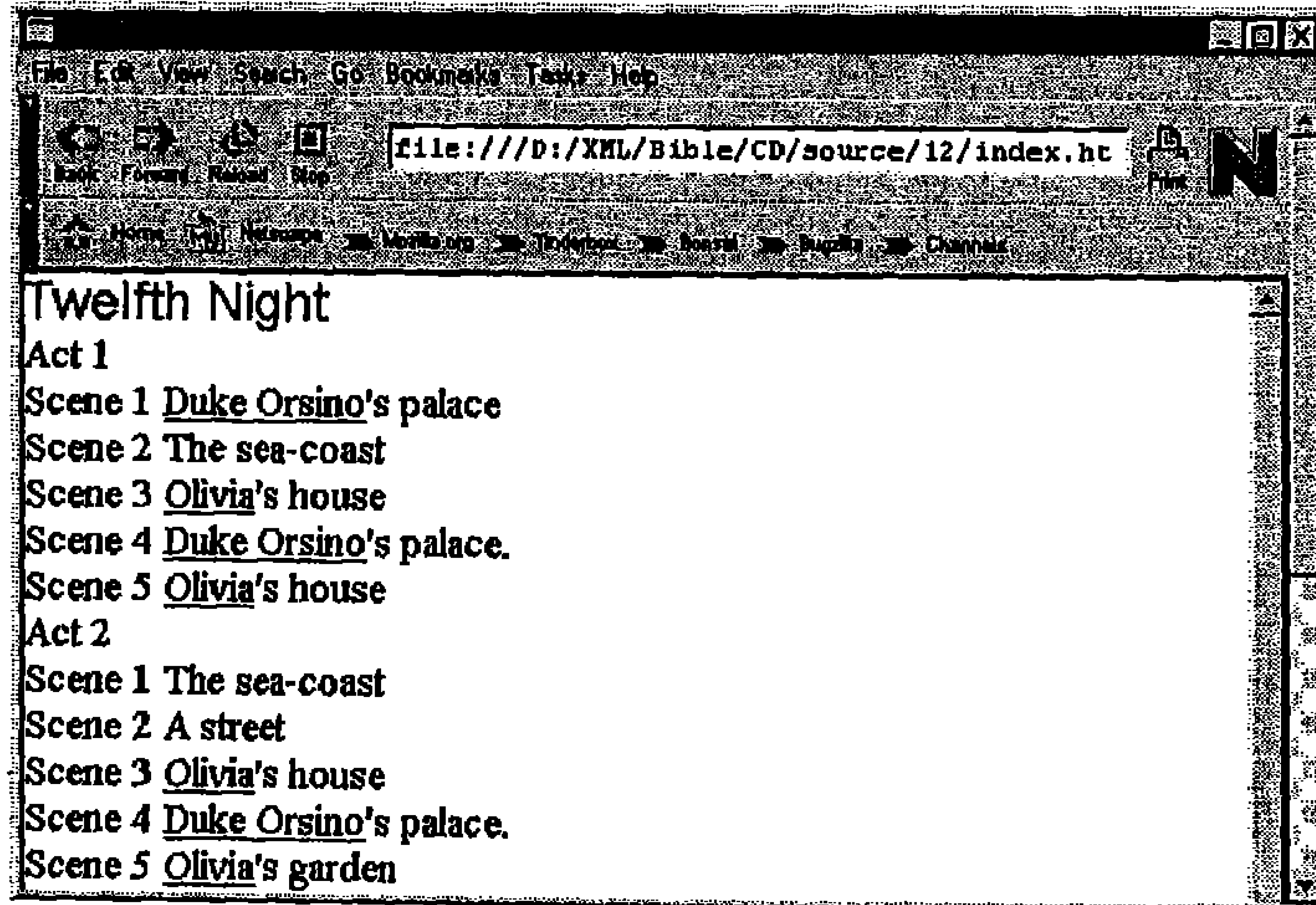
ولا تتوافق الإعدادات الافتراضية لهذه الصفات مع ما يقابلها باستثناء none. فمثلاً يمكنك أن تحدد أن يوضع خط تحت الفقرة أو يوضع فوقها خط أو أن توجد بينها فواصل أو تصبح وامضة "وإن كنت لا أفضل أن تفعل ذلك".



ولا يفرض على المستعرضات احترام النص الوامض وهو أفضل.

ومثلاً تحدد القاعدة التالية أن يوضع خط تحت CHARACTER للعنصر. ويوضح الشكل ١٢-٢١ النتائج على ملخص مسرحية الليلة الثانية عشر المذكورة في الشكل ١٢-٧.

```
CHARACTER { text-decoration: underline }
```



الشكل ١٢-٢١ ملخص مسرحية الليلة الثانية عشر بوجود خط تحت الشخصيات.

خاصية vertical-align

تحدد خاصية vertical-align كيف يتحدد وضع عنصر محول بالنسبة للعنصر الأساسي في النص والقيم المتوفرة هي:

baseline
sub
super
top
text-top
middle
bottom
text-bottom

ويمكنك أيضاً أن تستخدم النسبة المئوية لارتفاع السطر للعنصر والوضع الافتراضي هو baseline والذي يحاذي بين خط القاعدة للعنصر وخط القاعدة لعنصره الأساسي.

والقيم sub تجعل العنصر إلى رموز سفلي. أما قيمة text-top فهي تحاذي قمة العنصر مع عنصره الأساسي. وتقوم قيمة middle بمحاذاة نقطة المنتصف للعنصر مع خط القاعدة لعنصره الأساسي مع إضافة نصف ارتفاع x. وتقوم قيمة text-bottom بمحاذاة أسفل العنصر مع أسفل حجم الخط لعنصره الأساسي. أما قيمة top فتقوم بمحاذاة النص بأطول الحروف أو العناصر في السطر. وتقوم القيمة bottom بمحاذاة القيم مع أقصر الحروف أو العناصر في السطر. وتتغير المحاذاة بتغير أطول الحروف أو أقصرها.

فمثلاً قاعدة الملحوظات أسفل الصفحة قد تبدو مثل تلك التي تصف الحرف وتقوم بتصغير حجم الخط بنسبة ٢٠%.

```
FOOTNOTE_NUMBER { vertical-align: super; font-size: 80% }
```

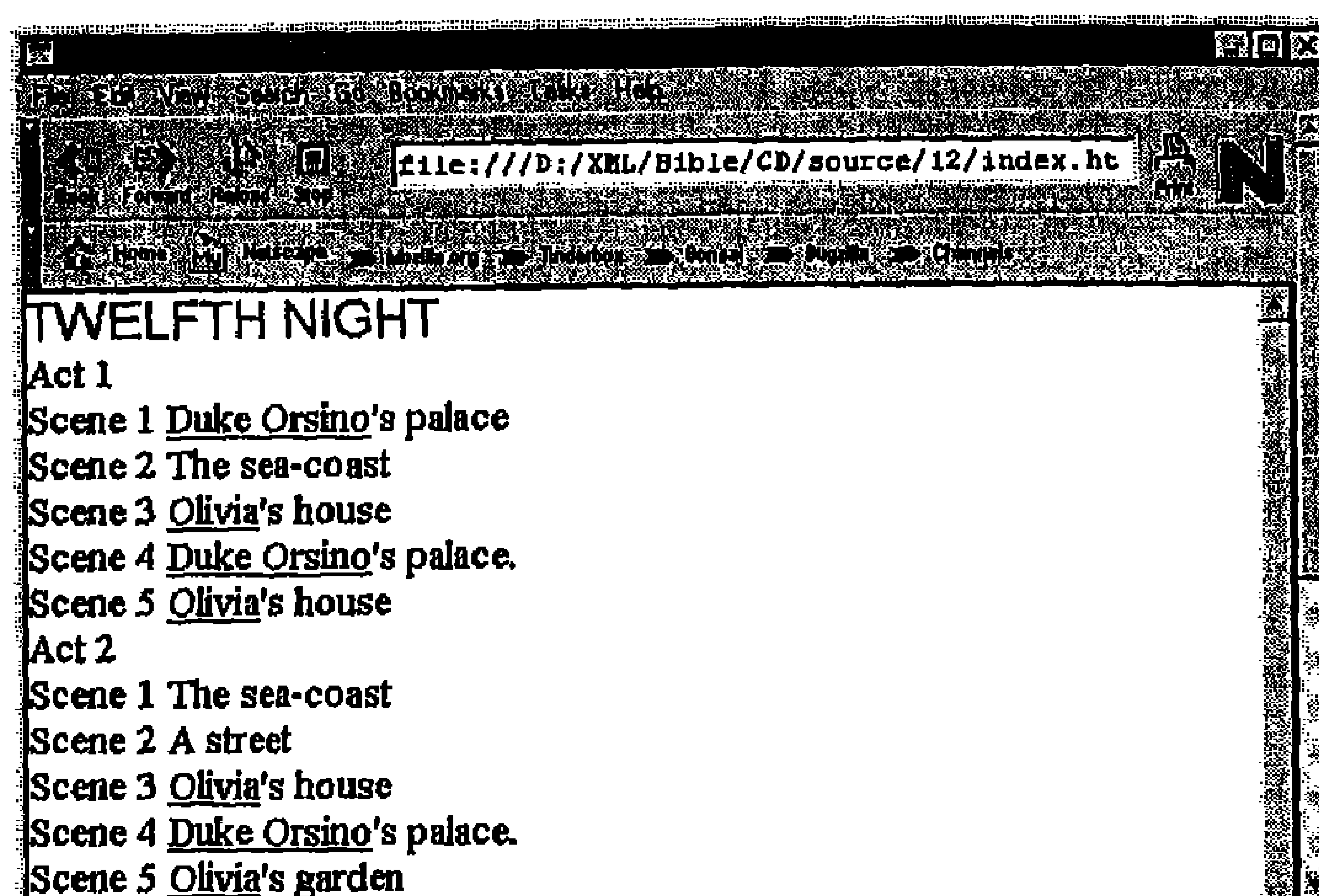
خاصية text-transform

تتيح لك خاصية text-transform أن تجعل النص كله بالحروف الكبيرة أو كله بالحروف الصغيرة أو أن تجعل الحروف الأولى كبيرة وهي مفيدة جداً في حالة العناوين مثلاً والقيم المتاحة هي:

capitalize
uppercase
lowercase
none

أما جعل الخط بحروف كبيرة يجعل كل الحروف تكتب كبيرة مثل هذه الجملـــة PLACING THE SENTENCE IN UPPERCASE, HOWEVER, MAKES EVERY LETTER IN THE SENTENCE UPPERCASE. والقاعدة التالية تقوم بتحويل عنصر TITLE إلى حروف كبـــيرة في ملخص الليلة الثانية عشر ويوضح شكل ١٢-٢٢ الملخص بعض إضافة هذه القاعدة.

```
TITLE { text-transform: uppercase }
```



الشكل ١٢-٢٢ TITLE من الملخص أصبح بحروف كبيرة.



تعتمد خاصية text-transform على اللغة إلى حد ما حيـث أن هنـاك بعـض اللغات مثل الصينية مثلاً ليست لديها أي حروف كبيرة وأخرى صغيرة.

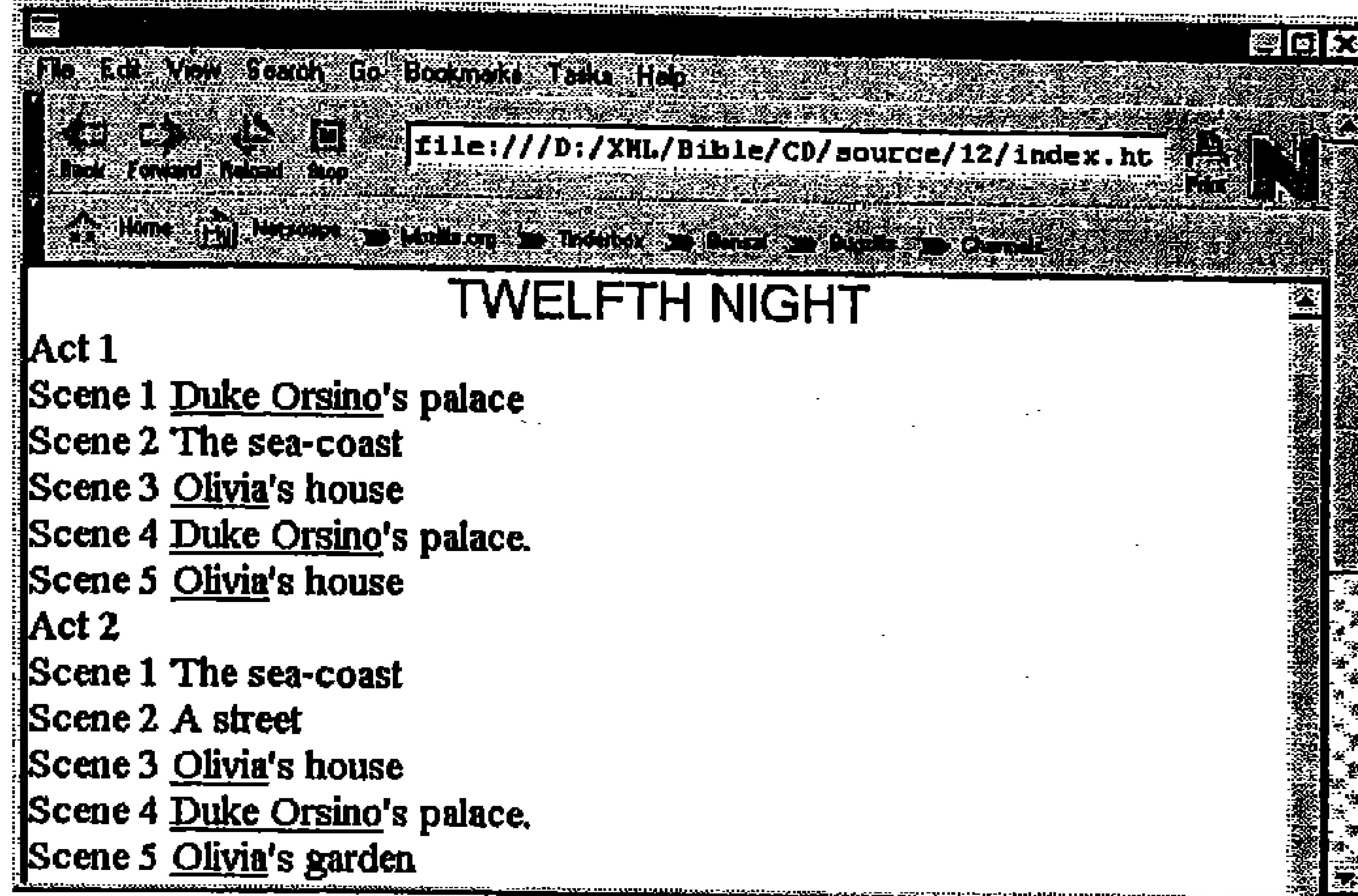
خاصية text-align

تطبق خاصية text-align على عناصر التجمع فقط وهي تحدد ما إذا كان النص ككـل سـيتم محاذاته مع الجانب الأيمن أم الجانب الأيسر أو سيكون في المنتصف أو يتساوى من الجـانبين. والقيم المتاحة هي كما يلي:

left
right
center
justify

والقاعدة التالية تجعل عنصر TITLE في المنتصف في ملخص مسرحية الليلة الثانية عشـر وتتساوى جميع العناصر الأخرى. ويوضح الشكل ١٢-٢٣ الملخص بعد إضافة هذه القاعدة:

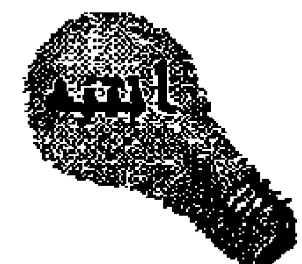
```
TITLE { text-align: center }
SYNOPSIS { text-align: justify }
```



الشكل ١٢-٢٣ عنصر TITLE في المنتصف وباقي العناصر متساوية.

خاصية text-indent

تطبق خاصية text-indent على العناصر في مستوى التجمع فقط وتحدد مستوى ارتفاع السطر الأول للمجموعة وتحدد المجموعة كلها بناءاً على ذلك وتقوم إعطاؤه إما قيمة مطلقة أو قيمة نسبة مئوية من العرض للعنصر الأساسي وقد ينتج تدلي الحروف إذا كانت القيمة سالبة.



ولكي تقوم بمساواة كل الأسطر وليس السطر الأول فقط تستخدم مربع الخصائص الذي ستتم مناقشته في القسم التالي لكي تعد حافة إضافية من الناحيـــــة اليســـرى للعنصر.

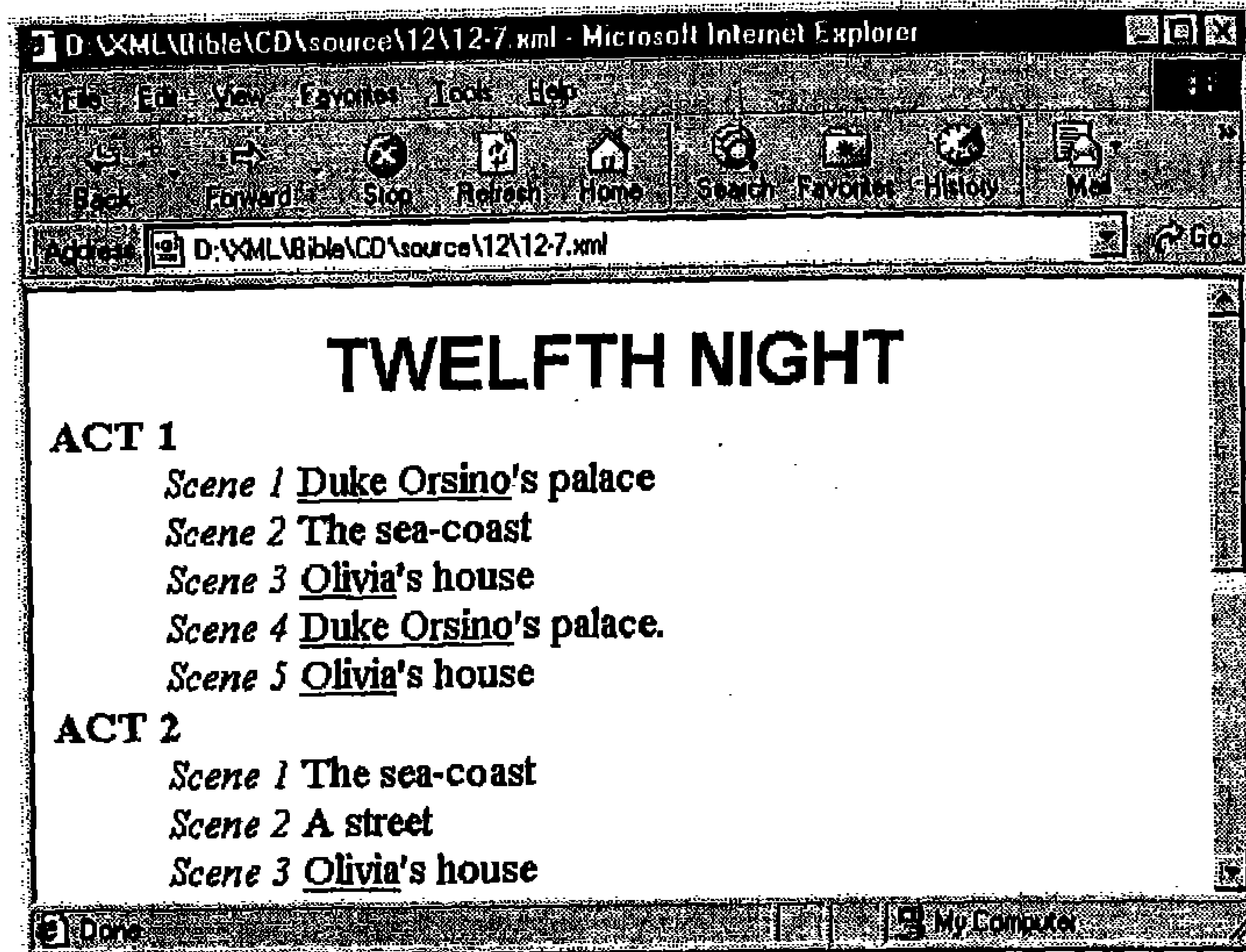
ومثلاً القاعدة التالية تجعل المشاهد يدخل للداخل بمقدار نصف بوصة ويوضح الشـــكل ١٢-٢٤ الملخص بعد إضافة هذه القاعدة:

```
SCENE { text-indent: 0.5in }
```

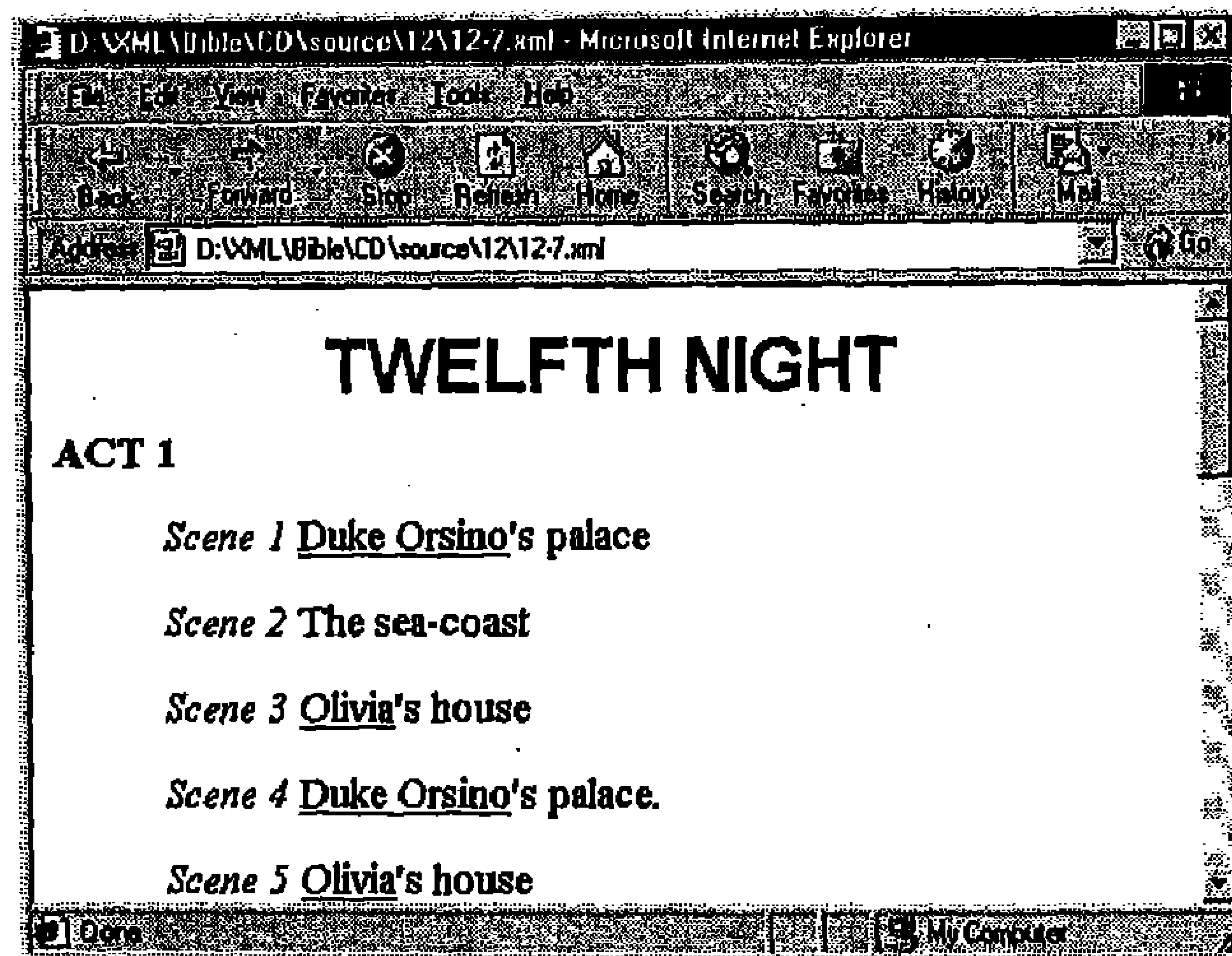
خاصية line-height

تحدد هذه الخاصية المسافة بين الخطوط الأساسية لعدة سطور متتالية ويمكن أن تعطـــي كرقـــم مطلق أو طول مطلق أو كنسبة مئوية من حجم الخط فمثلاً تضاعف القاعدة التالية بين مســـافات عنصر SYNOPSIS ويوضح الشكل ١٢-٢٥ ملخص المسرحية بعد إضافة هذه القاعدة:

```
SYNOPSIS { line-height: 200% }
```



الشكل ١٢-٢٤ عنصر SCENE مع توابعه في الملخص وكلـــها تدخل بمقدار نصف بوصة.



الشكل ١٢-٢٥ ملخص به مسافات مضاعفة.

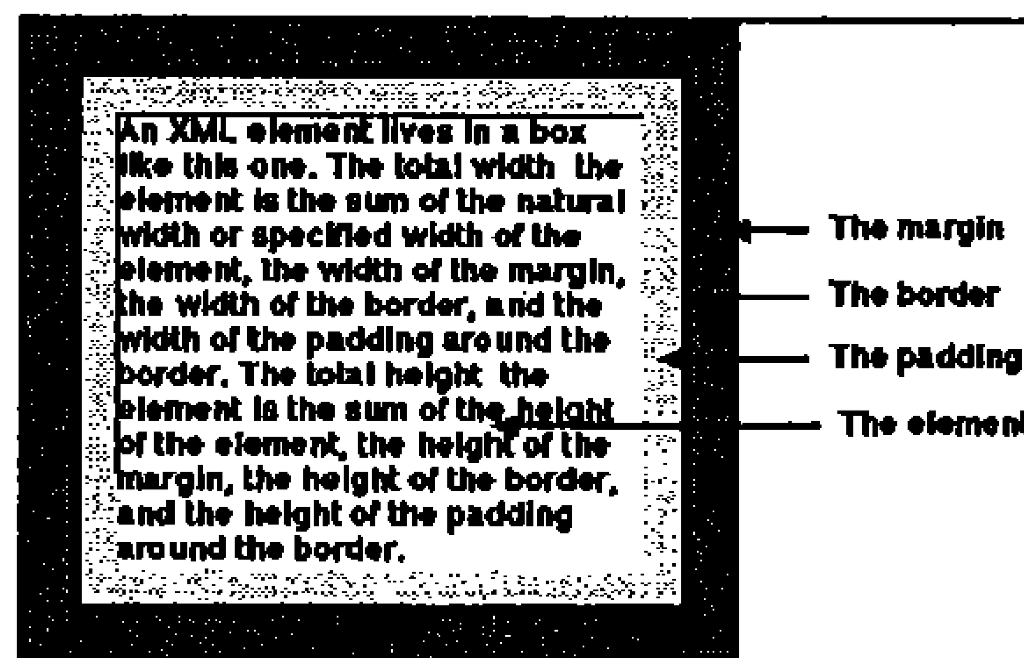
ولا تبدو المسافات المضاعفة جذابة على وجه الخصوص لذلك فسوف أزيلها. ويتم إضافـــة بعض الحواف الإضافية حول العناصر المنفصلة في الجزء التـــالي وتلخـــص القائمـــة ١٢-١٥ الإضافات التي تمت إضافتها خلال هذا الجزء على ملخص ورقة الرمز "مـــا عـــدا المســـافات المضاعفة".

تعليمات البرمجة ١٢-١٥: ورقة رمز الملخص وبها خصائص النص

```
SYNOPSIS, TITLE, ACT, SCENE { display: block }
SCENE_NUMBER { font: italic smaller serif }
TITLE { font: bold x-large Helvetica, sans-serif }
SYNOPSIS { font: 14pt Times, "Times New Roman", serif }
ACT_NUMBER            { font-variant: small-caps}
ACT_NUMBER:first-letter { font-variant: normal}
ACT_NUMBER { font-weight: bold}
CHARACTER { text-decoration: underline }
TITLE { text-transform: uppercase }
TITLE { text-align: center }
SYNOPSIS { text-align: justify }
SCENE { text-indent: 0.5in }
```

خصائص Box

تصف CSS قنب ذو بعدين يتم رسم النتائج فوقه ويتم وضع هذه النتائج في مستطيلات خيالية تسمى المربعات "الصناديق" وتتم محاذاة حوافها متوازية مع حواف القنب. وتتيح لك خاصية Box أن تحدد عرض وارتفاع الحواف والحواشي والإطارات وأحجام وأوضاع المربعات كل على حدة. ويوضح الشكل ١٢-٢٦ كيف تتعلق هذه الخصائص ببعضها البعض.



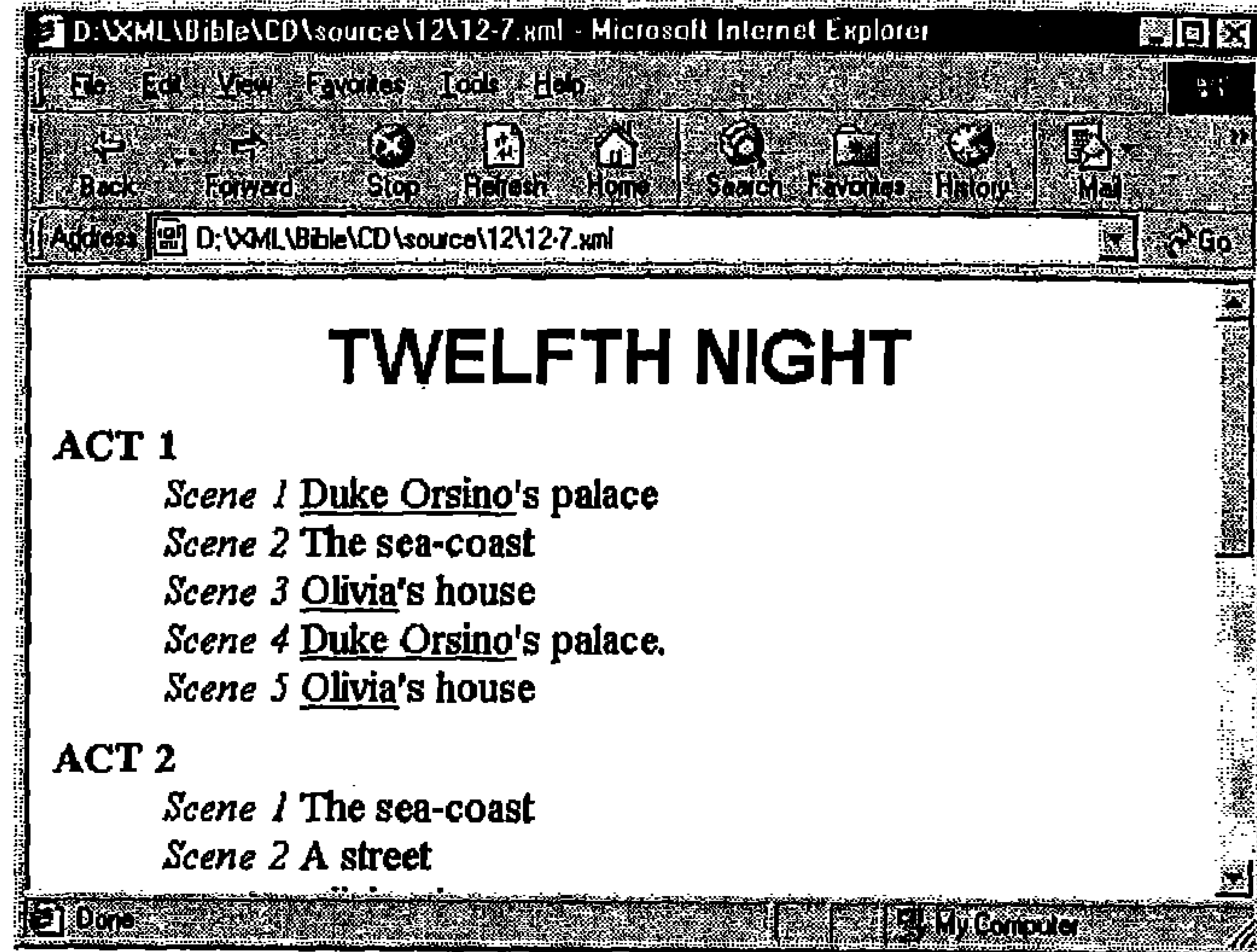
الشكل ١٢-٢٦ مربع CSS وبه الحواف والإطارات والحشو.

خواص Margin

تحدد خواص Margin مقدار المسافة التي تتم إضافتها للمربع خارج الإطار. ويمكن أن تعد هذه الخصائص لكل من الجزء العلوي والسفلي واليمين أو اليسار كل على حدة باستخدام خصائص margin-top و margin-bottom و margin-right و margin-left ويمكن أن يعطى لكل

حافة طول مطلق أو نسب مئوية من حجم عرض العنصر الأساسي. فمثلاً يمكنـــك أن تضيـف مسافة قليلة بين كل عنصر ACT والعنصر السابق له وذلك بتحديد خاصية ACT's margin-top على 1ex كما في القاعدة التالية كما يوضح الشكل ١٢-٢٧.

```
ACT { margin-top: 1ex }
```



الشكل ١٢-٢٧ الحافة العليا لعنصر ACT أكبر.

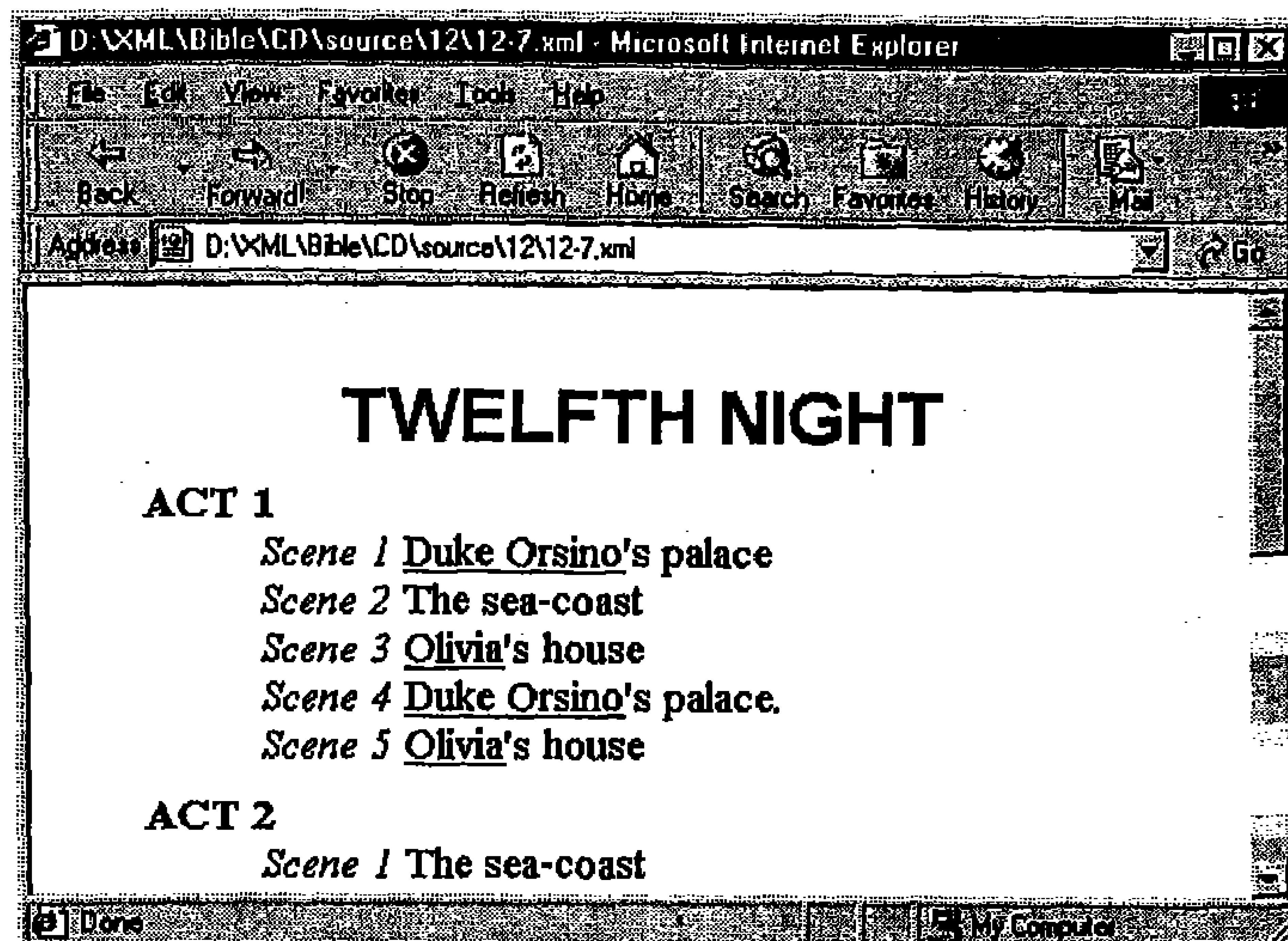
ويمكنك أيضاً إعداد الحواف الأربعة مرة واحدة وذلك باستخدام خاصية اختصـــار margin فمثلاً يمكنك أن تضيف مساحة بيضاء إضافية حول كل مستند مسرحية الليلة الثانية عشر وذلــك بتحديد خاصية margin على العنصر في المستوى الجذري "وهو SYNOPSIS في هذا المثـــال" كما هو موضح في القاعدة التالية التي يوضحها الشكل ١٢-٢٨.

```
SYNOPSIS { margin: 1cm 1cm 1cm 1cm }
```

وفي الحقيقة فأن هذا مثل استخدام قيمة أحادية للحافة والتي تترجمـــها CSS وتطبقـــها علـــى الحواف الأربعة.

```
SYNOPSIS { margin: 1cm }
```

وتعطى قيمتين لخاصية margin بحيث تطبق الأولى على الحافة العليا والسفلي وتطبق الثانية على الحافتين اليمنى واليسرى. وفي حالة وجود ثلاث قيم لخاصية margin تطبق القيمة الأولـــى على الحافة العليا والثانية على الحافتين اليسرى واليمني بينما تطبق الثالثة على الحافة الســـفلي. وربما من الأســـهل اســـتخدام خصـــائص margin-top وmargin-bottom وmargin-left منفصلة.



الشكل ١٢-٢٨ مساحة بيضاء سنتيمتر واحد حول الملخص كله.

خصائص Border

معظم المربعات ليست لها الإطارات وهي مربعات تخيلية تؤثر على تخطيط المحتويات ولكنها لا يراها القارئ كمربعات ومع ذلك يمكنك أن تجعلها مرئية بأن ترسم خطاً حولها لاستخدام خصائص border. وتسمح لك هذه الخواص أن تحدد النمط والعرض واللون لكل إطار.

نمط الحد

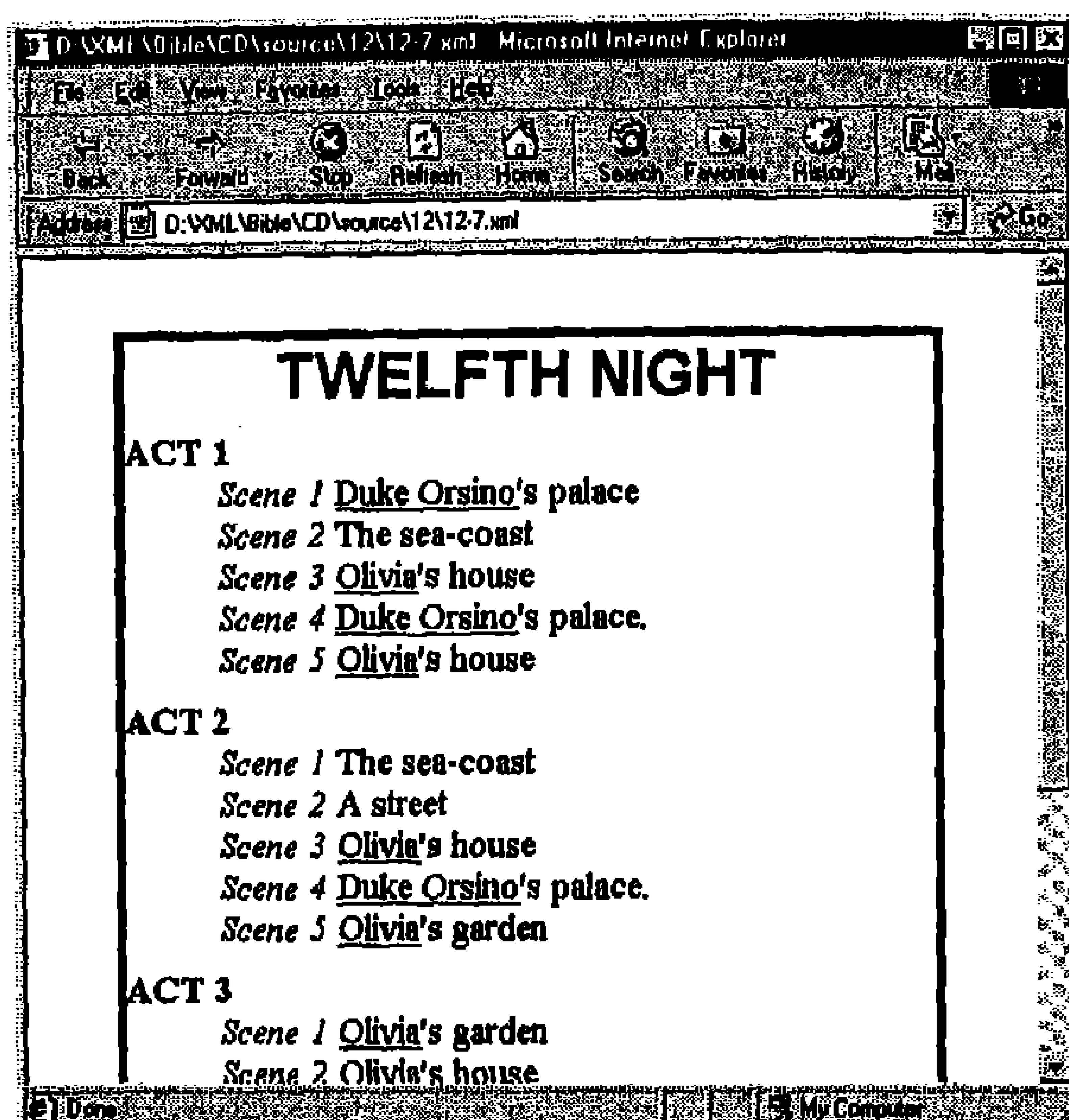
ولا يتم رسم الحدود حول المربعات افتراضا بصرف النظر عن عرض ولون الحد. ولكي تجعله مرئياً يجب تغيير خاصية border-style للمربع من القيمة الافتراضية none لأحد القيم التالية:

dotted
dashed
solid
double
groove
ridge
inset
outset

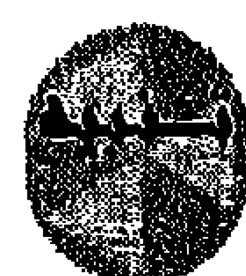
ويمكن أن يكون لخاصية border-style من واحد إلى أربعة قيم ومثل خاصية margin تطبق القيمة الواحدة على الأربعة حدود. وعند الإعداد الثنائي تحدد الحافتين العليا والسفلي على النمط الأول والحافتين اليسرى واليمنى على النمط الثاني. وفي حالة الإعداد الثلاثي تحدد الحواف

العليا ويمني واليسرى ثم السفلي وفق هذا الترتيب. أما الإعداد الرباعي للقيم فيحدد الحواف الأربعة ومثلاً تحيط القاعدة التالية بعنصر SYNOPSIS بحواف داكنة ويوضح شكل ١٢-٢٩ النتائج على Explorer 5.0. في هذه الحالة يكون للحدود التأثير الثانوي في جعل الحواف أكثر وضوحاً "وتذكر أن الحواف خارج الحدود".

```
SYNOPSIS { border-style: solid }
```



الشكل ١٢-٢٩ حدود حول الملخص.



يمكن لـ Explorer 5.0 أن تعرض الحدود الخالية من الزركشة فقط وكل الأنماط الأخرى تظهر بشكل حدود بسيطة خالية من أي أشكال.

عرض الحدود

هناك أربعة خواص عروض متاحة للحدود والتي تحدد عرض الحد من الجهة العليا والسفلي واليمني واليسرى للمربع وهي كما يلي:

١- border-top-width.

٢- border-right-width.

٣- border-bottom-width.

٤- border-left-width.

وكل منها يمكن أن تحدد كطول مطلق أو كأحد الكلمات الأساسية وهي thin وmedium وthick ولا يمكن أن يكون العرض سالباً ولكن يمكن أن يكون صفراً.

فمثلاً لكي تضم عنصر SYNOPSIS في حد عرضه واحد باكسيل "وهو أقل عرض للحدود يمكن أن يعرضه أي كمبيوتر" فيمكنك استخدام القاعدة التالية لتحديد الخصائص الأربعة.

```
SYNOPSIS { border-style:       solid;
          border-top-width:    1px;
          border-right-width:  1px;
          border-bottom-width: 1px;
          border-left-width:   1px }
```

وإذا أردت أن تحدد نفس العرض لكل الحدود فمن الأسهل استخدام خاصية اختصار border-width ويمكن أن يكون لها من واحد إلى أربعة قيم. ويمكن إعداد قيمة واحدة للأربعة حدود أو قيمتين للحدين العلوي والسفلي فيتم إعدادهما على القيمة الأولى والقمة الثانية للحدين الأيمن والأسفل. وفي حالة استخدام ثلاثة قيم تستخدم القيمة الأولى للحد العلوي والقيمة الثانية للحدين الأيمن والأيسر والثالثة للعرض بهذا الترتيب. وفي حالة استخدام أربعة قيم تكون بالترتيب التالي الحد العلوي ثم الأيمن والأسفل وأخيراً الأيسر. ويتم ذلك وفق القاعدة التالية:

```
SYNOPSIS { border-style: solid; border-width: 1px }
```

لون الحد

يحدد إعداد خاصية border-color لون أحد الحدود أو لونها الأربعة معاً وتعد القيمة الواحدة الحدود الأربعة معاً أما إعداد قيمتين فيحدد الحدين العلوي والسفلي اللون الأول والحدين الأيسر والأيمن للقيمة الثانية وتحدد القيم الثلاثة الحدود العلوي والأيمن والأيسر بهذا الترتيب. وفي حالة القيم الأربعة تحدد فتحدد الحدود العلوي والأيمن والأسفل والأيسر بهذا الترتيب. والقيم السارية المفعول هي أية أسم لون أو ثلاثية RGB مثلاً لكي تضمن العنصر SYNOPSIS في حد عرضه واحد بكسل وذو لون أحمر فسوف تحتاج للقاعدة التالية:

```
SYNOPSIS { border-style: solid;
          border-width: 1px;
          border-color: red }
```

ولن أعرض الصورة هنا حيث أن هذا الكتاب أبيض وأسود فقط.

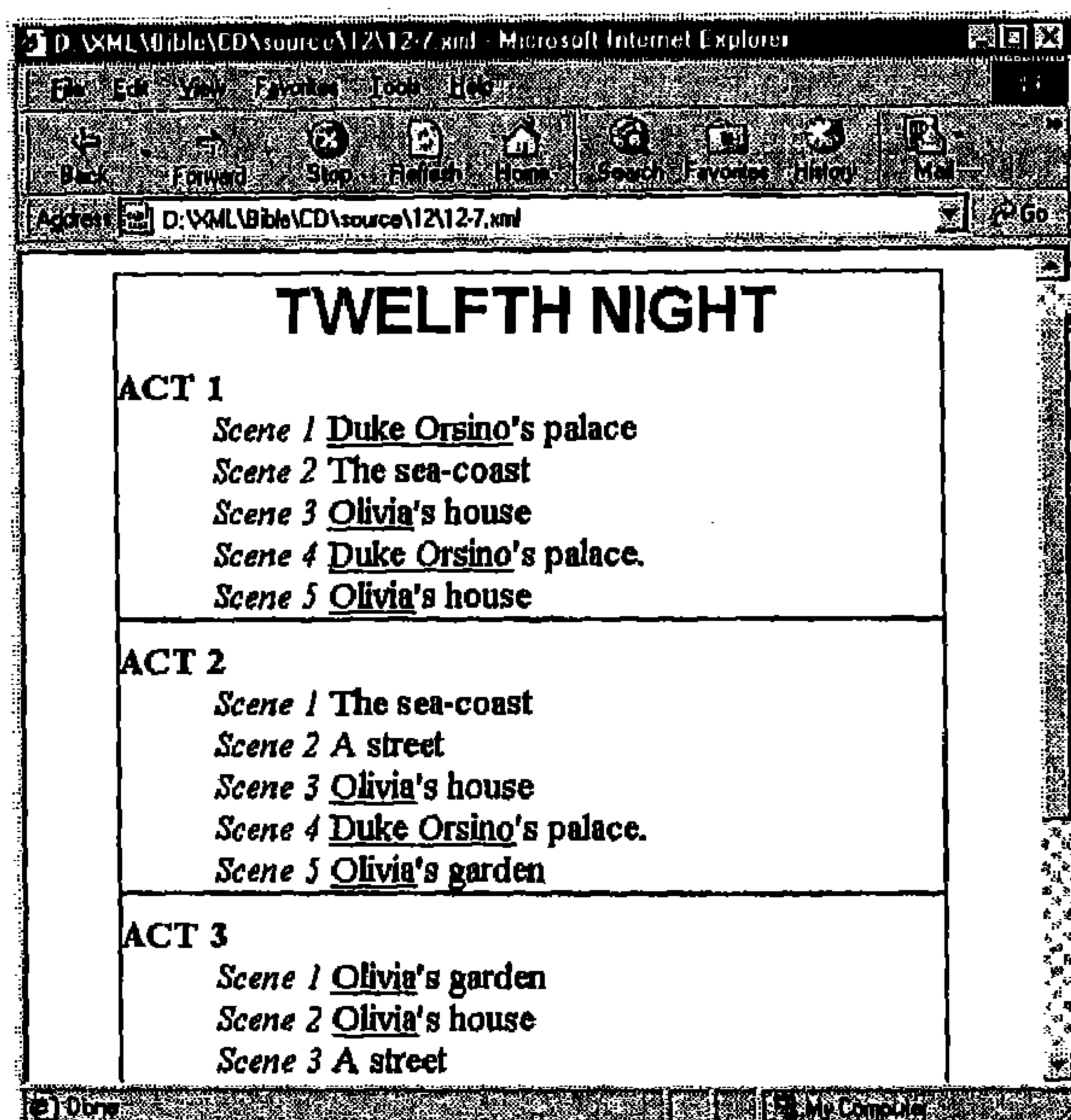
خواص اختصار Border

هناك خمسة خواص اختصار تتيح لك إعداد العرض والنمط واللون للحد باستخدام قاعدة واحـــدة وهي:

١- border-top
٢- border-right
٣- border-bottom
٤- border-left
٥- border

ومثلاً تمد خاصية border-top بالعرض والنمط واللون للحـــد العلـــوي وتقـــوم كـــل مـــن خصائص border-right و border-bottom و border-left بنفس العمل. والخصـائص الملغـاة يتم أعدادها وفق أعداد الخصـائص الأساسية لها. فمثلاً يوضح الشكل ١٢-٣٠ حد أزرق ســــاده عرضه ٢ باكسيل "وخط عرضي أسفل كل فصل من الفصول إذا أردت ذلك" ولكي تتوصل إلـــى ذلك عليك إتباع القاعدة التالية.

```
ACT { border-bottom: 2px solid blue }
```



الشكل ١٢-٣٠ حد سفلي أزرق بدون نقـــوش يشـــبه لعنصر HR الخاص بـــ HTML.

وتعد خاصية border الأربعة حدود وفق العرض والنمط والارتفاع فمثلاً القاعدة التالية تقوم برسم حد عرضه ٣ باكسيل أحمر دون نقش حول عنصر CHART:

```
CHART { border: 3pt solid red }
```

خواص Padding

تحدد خاصية Padding المساحة المتروكة داخل الحد الموجود في المربع. حيث أن حدود المربع إذا كانت ظاهرة تقع بين الحافة وبين الحشو. ويمكن إعداد الحشو لكل جهة على حدة لكـل مـن الحشة العليا والسفلي واليمني واليسرى باستخدام كل من خصـائص padding-top و padding-bottom و padding-right و padding-left ويمكن أن تعطي الحشو قيمة طول مطلقـة أو أن تعطي نسبة من حجم العنصر الأساسي لها فمثلاً يمكنك أن تلغي SYNOPSIS من الحد وذلك عن طريق إعداد خصائص الحشو الخاصة بها وفق القاعدة التالية:

```
SYNOPSIS { padding-bottom: 1em;
          padding-top: 1em;
          padding-right: 1em;
          padding-left: 1em }
```

وكذلك يمكنك إعداد الأربعة حدود مرة واحدة باستخدام خاصية اختصـار padding فمثـلاً تعبر القاعدة التالية نفس ما تقدمه القاعدة السابقة:

```
SYNOPSIS { padding: 1em 1em 1em 1em }
```

وفي الحقيقة فإن النتيجة تماثل أستخدام قيمة أحادية لخاصية الحشـو والتـي تترجمـها CSS وكأنك تطبقها على الأربعة جوانب:

```
SYNOPSIS { padding: 1em }
```

وفي حالة وجود قيمتين فإن الأولى يتم تطبيقها على الجهتين العليا والسفلي ويتم تطبيق الثانية على اليمين واليسار. وفي حالة وجود ثلاثة قيم للحشو فإنها تستخدم فإن الأولي تسـتخدم الجهـة العليا وتستخدم الثانية للجهة اليمنى واليسرى بينما تستخدم الثالثة للجهة السفلي. وقد يكـون مـن الأسهل.

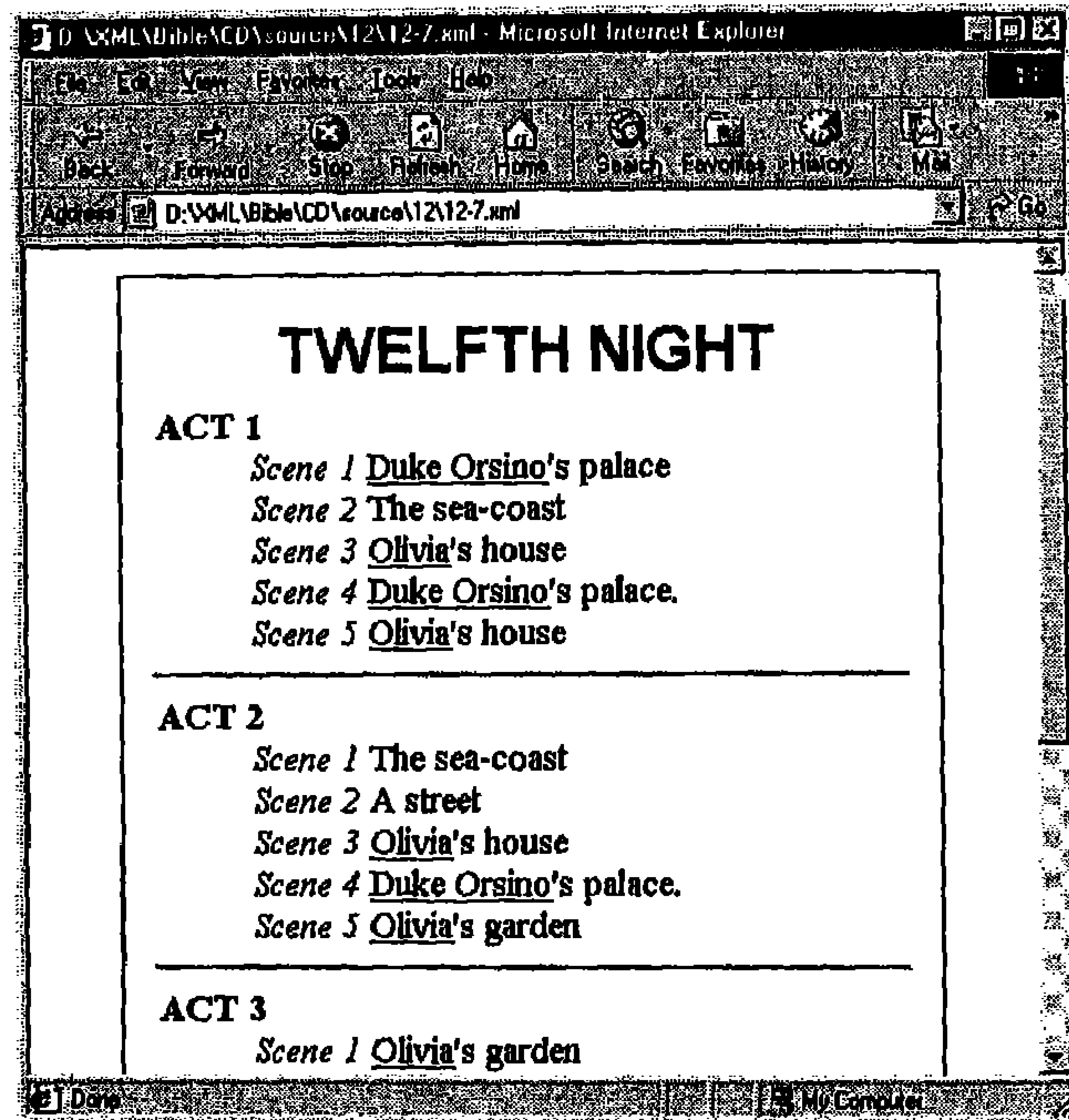
استخدام الخصـائص المنفصلـة padding-top و padding-bottom و padding-right و padding-left. وبما أن الحدود الموجودة بعد كل فصل تبدو قريبة بعد الشيء فلنضف المزيـد من مساحة الحشو بين نهاية الفصل والحد وذلك باستخدام خاصية padding-bottom ويوضـح الشكل ١٢-٣١ النتائج وعموماً فإنها فكرة جيدة أن تستخدم القليل من الحشو حول الحدود لكـي تجعل النص أسهل في القراءة.

```
ACT { padding-bottom: 1ex }
```

خصائص Size

يمكن أن يفرض حجماً معيناً على المربع وذلك بتطبيق خاصتي width و height وستتم موازنـة محتويات المربع حتى تلائم الحجم. وبالرغم من أنه يمكنك عمل ذلك مع مربعات النص إلا أنـــه من الأكثر شيوعاً في الاستخدام مع العناصر التي يتم استبدالها مثل الصور. ويمكـــن أن تكـون قيمتي العرض والارتفاع مطلقتين أو كنسبة مئوية من العنصر الأساسي أو ككلمك أساسية مثـــل auto "وهو الوضع الافتراضي" ولكي تقوم بتوجيه المستعرض عليك استخدام الحجم الأصلـــي. مثلاً تحاول القاعدة التالية أن تضع عنصر SYNOPSIS بأكمله داخل مربع يتكـــون مـــن ٣×٣ بوصة.

```
SYNOPSIS { padding: 1em; width: 3in; height: 3in }
```



الشكل ١٢-٣١ يخفف الحشو من وقع منظر الحدود علـــى العين.

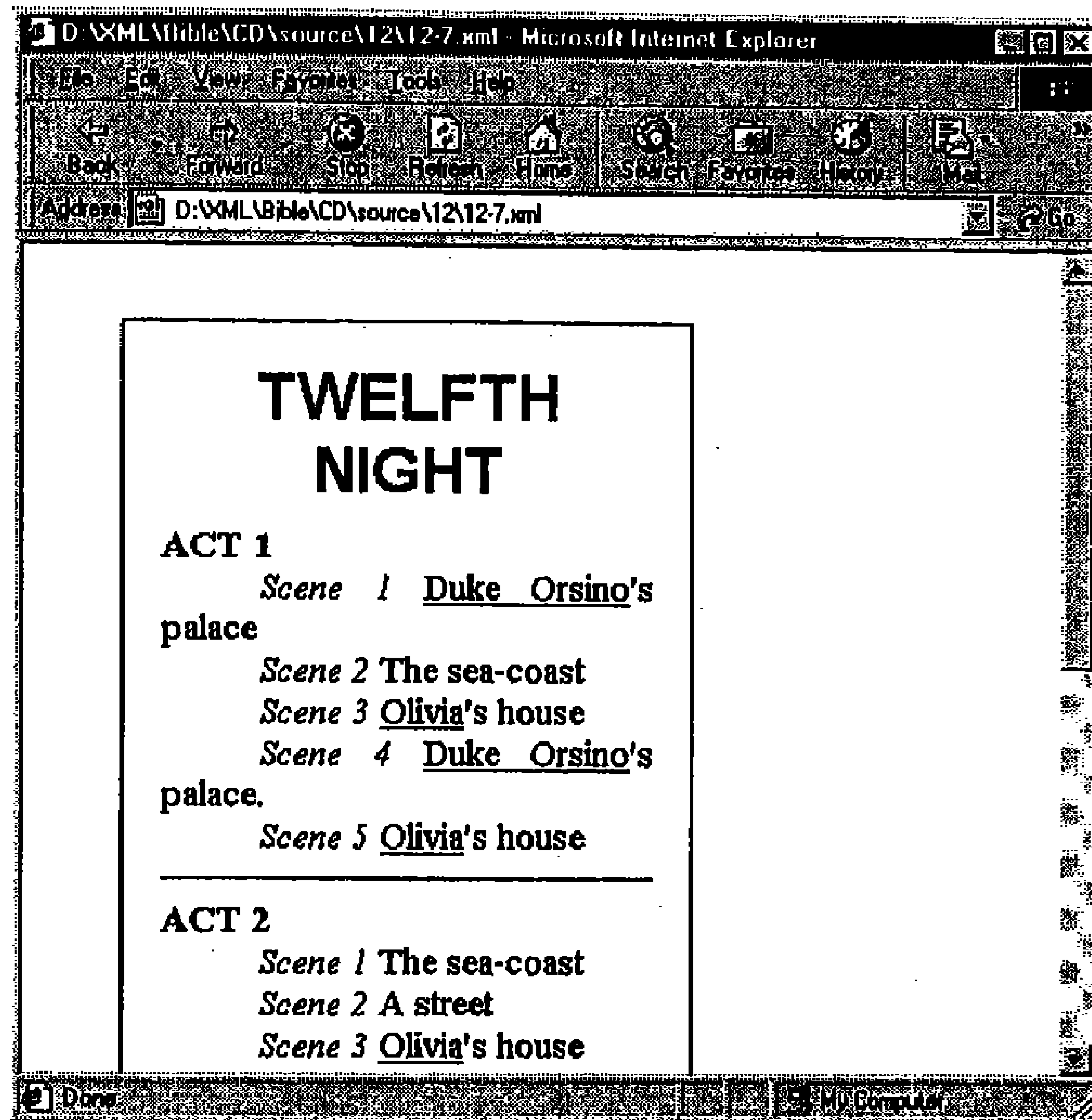
يوضح الشكل ١٢-٣١ النتائج كما تعرض على Internet Explorer 5.0. وعندما يواجـهك عنصر حجمه أكبر مما يسمح به مربعه فأن Internet Explorer يحدد العرض ويزيد الارتفاع. بينما تسمح Mozilla للنص أن ينساب خارج المربع وقد يغطـــي العنـــاصر الموجـــودة أســـفله. وتتعامل المستعرضـات مع النصوص التي لا تستطيع المربعات احتوائها بأساليب مختلفـــو وقـــد تكون غير متوقعة.

ولذلك فإنه يمكن تحديد الحجم بالتحديد من خـــلال تصميـــم الويـــب الـــذي يناســـب معظـــم المستعرضات.

وإذا تم إعداد width لكي يكون مطلقاً أو لوحدات نسبية وتم إعداد height على auto فـــإن الارتفاع سيتغير بما يتناسب مع العرض.

خصائص Positioning

تتبع العناصر التجميعية الموجودة داخل نفس العنصر الأساسي تتبع بعضها البعض فـــــي نفـــس الصفحة افتراضيا ولا تصطف بجوار بعضها أو تلتف حول بعضها البعض ولكن يمكنك تغيـــير ذلك باستخدام خصائص float وclear.



الشكل ١٢-٣٢ الملخص ذو ٣ بوصات ارتفاع و٣ بوصـــات عرض كما تظهره Mozilla.

خاصية float

يمكن أن تعد خاصية float والتي قيمتها الافتراضية none من ناحية left أو right. فإذا كـــانت القيمة left فإنه سيتحرك العنصر إلى الناحية اليسرى من الصفحة ويحيطه النـــص مـــن جهـــة اليمين. وفي حالة HTML سيتصرف IMG مع ALIGN="LEFT" بنفس الأسلوب. أما إذا كـــانت

القيمة right فأن العنصر سيتحرك إلى يمين الصفحة ويلتف حوله النص من ناحية اليسار وفـــي حالة HTML هكذا بفعل IMG مع "ALIGN="RIGHT.

وليست هناك طريقة قياسية لتثبيت الصور في ملفات XML لذلك سنستخدم صورة غير حقيقية في الخلفية هذا المثال مع استخدام بعض خصائص CSS. وفي القائمة ١٢-١٦ دعوة الحفل مـــع إجراء بعض التعديلات وبها عنصر IMAGE خالي أما القائمة ١٢-١٧ فهي ورقة نمـــط التـــي تحدد خلفية ملف party.gif كخلفية لعنصر IMAGE وهي تحدد كل مـــن العـــرض والارتفـــاع لخصائص IMAGE وكذلك تحدد float إلى left. والشكل ١٢-٢٣ يوضح النتائج.

تعليمات البرمجة ١٢-١٦ : دعوة الحفل وبها عنصر IMAGE خالي

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="12-17.css"?>
<INVITATION>
  <IMAGE />
  <TEXT>
  You're invited to a party on December 31, 1999 to celebrate
  the new millennium! You're invited to a party on December 31,
  1999 to celebrate the new millennium! You're invited to a
  party on December 31, 1999 to celebrate the new millennium!
  You're invited to a party on December 31, 1999 to celebrate
  the new millennium! You're invited to a party on December 31,
  1999 to celebrate the new millennium! You're invited to a
  party on December 31, 1999 to celebrate the new millennium!
  You're invited to a party on December 31, 1999 to celebrate
  the new millennium! You're invited to a party on December 31,
  1999 to celebrate the new millennium! You're invited to a
  party on December 31, 1999 to celebrate the new millennium!
  You're invited to a party on December 31, 1999 to celebrate
  the new millennium! You're invited to a party on December 31,
  1999 to celebrate the new millennium!
  </TEXT>
</INVITATION>
```

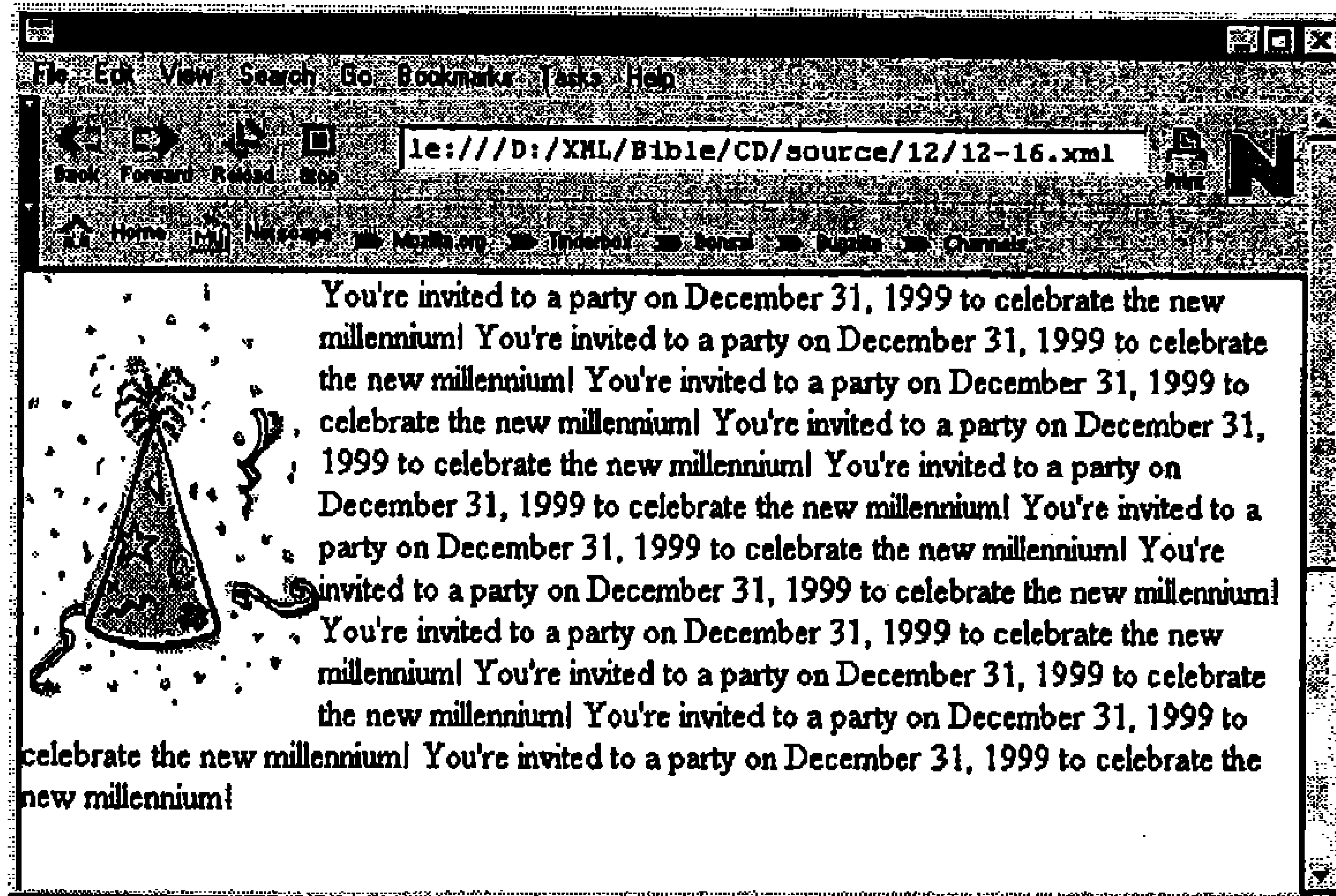
تعليمات البرمجة ١٢-١٧: ورقة النمط التي تحمل IMAGE

```
INVITATION { display:block; }
IMAGE { background: url(party.gif) no-repeat center center;
    width: 134px;
    height: 196px;
    float: left; }
TEXT { display: block }
```

الخاصية clear

تخصص الخاصية clear إذا ما كان العنصر ستكون له عناصر تطفو بجانبه. وإذا لم يكن له فإن العنصر سينقل تحت أي عنصر طفو يسبقه وهذا يقارب عنصـــر <BR CLEAR="ALL"> فـــي HTML والقيم المتاحة هي:

none	right
left	both

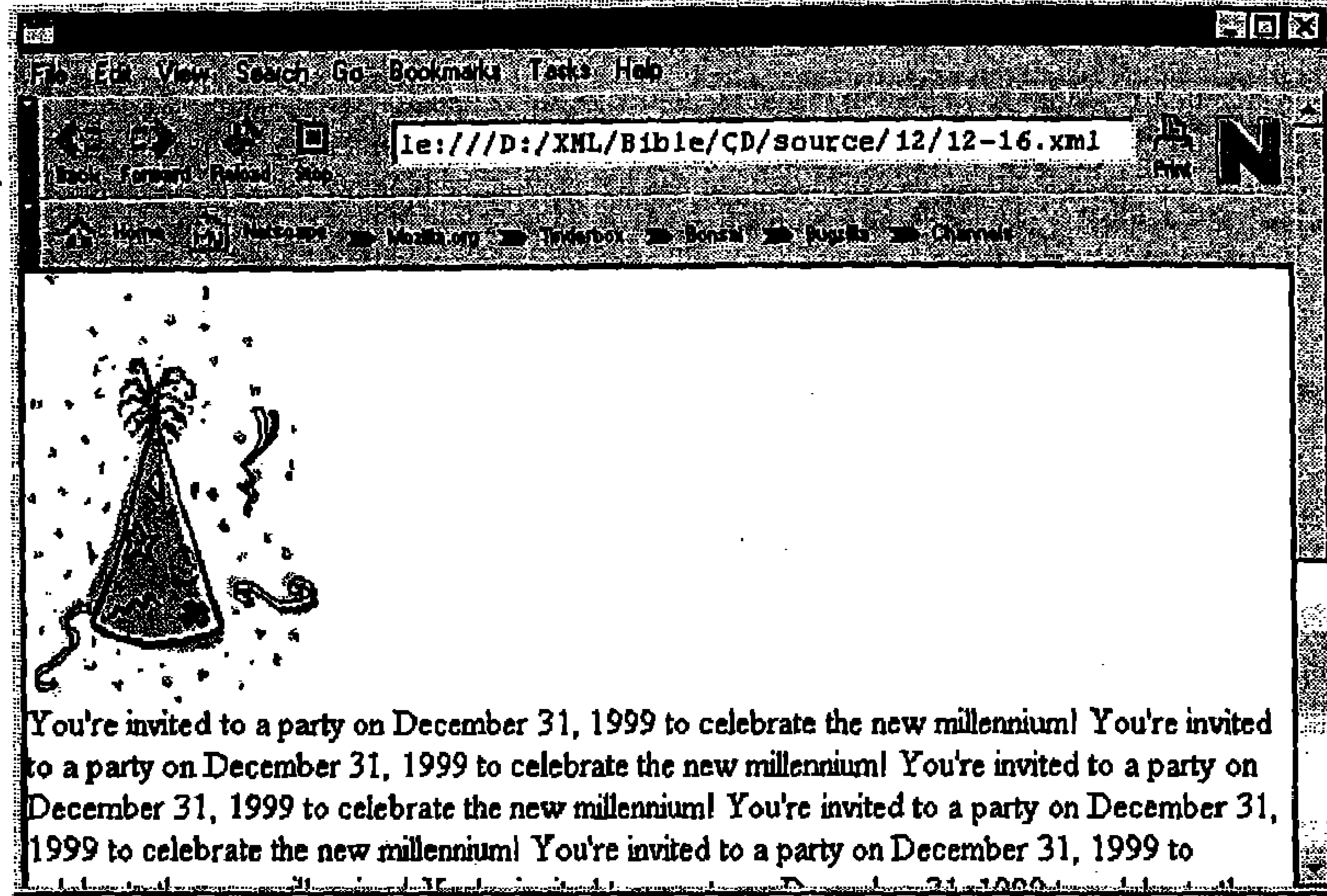


الشكل ١٢-٣٣ صورة دعوة الحفل تطفو ناحية اليسار.

والوضع الافتراضي هو none ويتيح للعناصر الطهور على جهتين العنصر. أما القيمـة left فتمنع العناصر من الطفو في الجهة اليسرى من العنصر. وتمنع القيمة right العناصر من الطفـو على الجهة اليمنى من العنصر. وتمنع القيمة both من طفو العاصر من على جهتي العنصـــر. فمثلاً إذا قمت بإضافة القاعدة التالية لورقة النمط الموجودة في القائمة ١٢-١٧.

TEXT { clear: left }

فإنه بالرغم من أن العنصر IMAGE يود الطفو إلى يسار العنصـــر TEXT، فـــإن TEXT لا يسمح بذلك كما هو موضح في الشكل ١٢-٣٤. فإن لا IMAGE يزال على اليسار ولكن تم دفـــع TEXT إلى أسفل تحت الصورة.



الشكل ١٢-٣٤ صورة دعوة الحفل مع خاصية clear معدة ناحية اليسار.

خلاصة

تعلمت في هذا الفصل:

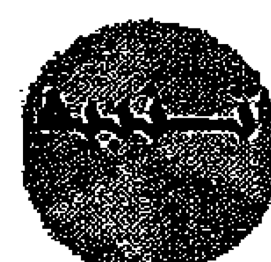
- أن لغة CSS هي لغة مباشرة لتطبيق الأنماط على محتويات العناصر التي تعمل جيداً مع HTML أو تعمل أفضل مع XML.
- المحدد هي قائمة ذات فواصل للعناصر التي يتم تطبيق القاعدة عليها.
- يمكن أن تطبق CSS القواعد على العنـــاصر ذات الأنـــواع المعينـــة ذات خصـــائص CLASS أو ID المحددة.
- يمكن توريث بعض "وليس كل" خصائص CSS بواسطة العناصر الفرعيـــة التـــي يتـــم التطبيق عليها.
- إذا تم تطبيق قاعدة متعددة على عنصر أحادى يتم تتالي خصائص التنسيق بما يتوافق مع ذلك.

- يمكنك تضمين ملحوظات مثل /* */ C-like في أوراق نمط CSS.
- يمكن تحديد الأطوال بنسب مطلقة أو نسبية كما تفضل.
- تحدد خاصية display ما إذا كان العنصر متجمع أو مصفوف أو في قائمة.
- تحدد خصائص الخط شكل الخط والنمط والحجم وثقل الخط للنص.
- يتم تحديد عناصر اللون بالمسافات إما بالديسيمل أو هيكساديسمل أو بالنسب المئوية.
- تتضمن خصائص الخلفية اللون والصورة ووضع الصورة وكذلك تغطية الصورة.
- تتيح لك خصائص Text من تعديل ارتفاع الخط والمسافات بين الكلمات والمسافات بين الأحرف وكذلك المحاذاة الأفقية والرأسية وكذلك تزيين الحروف وحجمها.
- تسمح لك خصائص Box أن تحدد الأوضاع النسبية والمسافات بين العناصر فوق الصفحة بالإضافة إلى التفاف الحواف حول العناصر.

هناك بعض الحدود لما يمكن أن تقوم به CSS Level 1. أولاً تتمكن CSS1 من ربط الأنماط بالمحتويات التي تظهر بالمستند فقط. ولا يمكنها أن تضيف محتويات إلى المستند ولا حتى أبسطها. علاوة على أنها لا تستطيع نقل المحتويات بأي طريقة مثل التصفية أو إعادة تنظيمها. ويمكن عمل ذلك باستخدام XSL أو Extensible Style Language. وتوفر CSS1 أقل مما تطلبه حتى فيما يتعلق بعمليات التنسيق المجردة. والأدهى من ذلك أنه لا توجد إمكانية عمل الجداول. وهناك العديد من العيوب الغير ظاهرة الأخرى. فهي لا يمكنها تناول النصوص التي تكتب من اليمين إلى اليسار مثل العبري والعربية أو النص الأفقي مثل اللغة الصينية التقليدية. وفي الفصل التالي سوف نقوم بالتعرف على CSS Level 2 والتي تتناول هذه العيوب وحدود CSS1.

أوراق نمط المستوى ٢

لقد تم نشر مواصفات أوراق المستوى ٢ "CSS2" للأنماط المتتالية بواسطة W3C في عام ١٩٩٨، متخطياً بذلك المستوى ١ CSS لجعل تنسيق مستندات XML وـHTML أكثر قوة بالطبع CSS2 يواجه نفس المشاكل مع HTML التي قد واجهتها CSS2. بينما، مع XML، يستطيع CSS2 تنسيق المحتويات في كلاً من الورق والويب غالباً مثل برنامج نشر على شاشة الكمبيوتر مثل PageMaker وQuark Xpress.



لم يتم تنفيذ معظم القواعد التي تمت مناقشها بواسطة المستعرضين العموميين. يجب أن يبدأ Mozilla في تنفيذ بعض هذه الأنماط، ولكن لم يتم التوصل بعد للتنفيذ الكلي.

ما هو الجديد في CSS2

يقوم CSS2 بدمج العديد من الميزات التي قد طالب بها كثيراً مصممي ومبرمجي الويب من بائعي المستعرضات تعد المواصفات في الحجم أكثر من ضعف CSS1، وهي لا تعد فقط تجميع للتغيرات والميزات الجديدة ولكن مسودة معادة للمواصفات الأصلية. هذا يجعل من هذه المواصفات مصدر فردي لجميع ورق الأنماط المتتالية، بناء الجملة والقواعد.

يستغرق البرنامج المعروف بعض الوقت لدعم جميع المواصفات الجديدة بأكملها، ولا يعد CSS2 استثنائياً. لأنك ستكتشف من خلال قراءتك لهذا الفصل أن كلاً من مكتشفي الإنترنت 5.0 وMozilla قد برعوا في تنفيذ تلك الخواص. أما المواصفات التي لم يتم تنفيذها قد تم كتابتها لكي تقتنع.

تمنحك الميزات الجديدة والعديدة لـ CSS2 القدرة على أن تحدد بدقة وتنسق العناصر التي توجد بمستندك. قد تمكنك العناصر المقترحة والفئات المقترحة الجديدة من تحديد التابع الأول لعنصر ما وضبط العنصر عندما يستقبل أي تركيز أو يتحكم في موضع العناصر الأخرى أوتوماتيكياً التي تدور حول تحديدات العنصر المحدد. تمكنك أنواع الوسائط من تطبيق مختلف الأنماط على المستندات التي قد تظهر في الوسائط المختلفة مثل الصفحات المطبوعة، شاشات الكمبيوتر، والإذاعات بالراديو. وقد تم تحسين الدعم من أجل الوسائط المرقمة مثـ[illegible]ـخ المطبوعة وعرض الشرائح درجة بدرجة تحكم في فاصل الصفحات أقوى بكثير. [illegible] الممكن الآن تنسيق العناصر في الجداول مثل الكتلة والمربعات المحولة. من الممكن ترقيم المسافة [illegible] للقوائم والتسلسلات أوتوماتيكياً. يتم توفير المزيد من الدعم للغات الغير عربية مثل اللغة [illegible] والصينية. وتستطيع للمرة الأولى تطبيق الأنماط السمعية التي تستطيع تعيين كيفية قراءة المستند ولا تستطيع تعيين كيفية معالجته. بالإضافة إلى ذلك، فإن CSS2 تستطيع تغيير تنفيذ بعض ميزات CSS1.

الفئات المقترحة الجديدة

تقوم الفئات المقترحة بتحديد العناصر التي تشترك في شيء ما ولكن ليس من الضروري أن تكون نفس النوع. فعلى سبيل المثال، hover الفئة المقترحة تُشير إلى أي عنصر يقف عليه المؤشر في ذلك الوقت، بغض النظر عن نوع العنصر. يعد CSS2 سبعة فئات. مقترحة جديدة. وهي مخططة بالتفصيل فيما يلي:

- first-child The :first: للفئة المقترحة تُحدد التابع الأول للعنصر.
- Focus The :focus: الفئة المُقترحة تُحدد الكائن الذي يكون عليه التركيز من لوحة المفاتيح.
- hover: The :hover: الفئة المُقترحة تُحدد الكائن المصمم وليس المنشط.
- lang: The :lang: الفئة المُقترحة تُحدد تلك العناصر المكتوبة بلغة محددة ومعرفة بسمة xml:lang.
- first: The :first: الفئة المقترحة تُحدد الصفحات اليسرى "بالطبع تلك هي الصفحات المرقمة زوجي" لمستند النسخة المطبوعة، وكأن مادة النسخ الصلبة سوف تدرج بالكتاب.
- right: The :right: الفئة المقترحة تُحدد الصفحات اليمنى "بالطبع تلك هي الصفحات المرقمة فردي" لمستند النسخة المطبوعة، وكأن مادة النسخ الصلبة ستصبح منضمة.

العناصر المُقترحة الجديدة

تقوم العناصر المُقترحة بتعريف عناصر محددة عن طريق المعلومات وليس عن طريق ما هو موجود بالفعل في إدخالات XML. على سبيل المثال، في CSS1 وCSS2، first-line: و-first letter يعتبروا عناصر مُقترحة تقوم بتحديد السطر الأول والحرف الأول لأي عنصر، حتى إذا كانوا غير ممثلين بالضرورة بأي عنصر.

يضيف CSS2 عنصرين. مُقترحين جديدين، after وbefore: يمكنك after pseudo-element من إدراج كائنات بعد عنصر المحدد. قد تكون تلك الكائنات صور، أجهزة عد أوتوماتيكية، أو نص. يمكنك before pseudo-element من إدراج كائنات قبل عنصر محدد. قد تكون تلك الكائنات أيضاً صور، أجهزة عد أوتوماتيكية، أو نص.

أنواع الوسائط

تعرف CSS2 عشرة أنواع من الوسائط يوجد بها معلومات مثل Braille، عروض كمبيوتر، حبر على ورق، والتلفزيون. تمكنك CSS2 من تحديد أنماط مختلفة للوسائط المختلفة على سبيل المثال، من الضروري استخدام خطوط كبيرة على شاشات الكمبيوتر ذات الدقة المنخفضة أقل من ١٢٠٠ من المطبوعات بوحدة قياس نقطة في البوصة.

الوسائط المرقمة

تمنحك CSS2 التحكم في فواصل الصفحات وطرق تعريف الصفحات الفردية في أي مستند، حتى يتمكن المصممين من تنسيق المستندات المطبوعة، دون المساس بمظهر المستند على الشاشة.

العالمية

بتوسع الإنترنت خارج نطاق العالم المتكلم باللغة الإنجليزية، قد تم إحراز المزيد من التقدم في دعم آلاف اللغات المنطوقة والمكتوبة حالياً أو في خلال التاريخ. كما يصف CSS2 المزيد من جمل اللغة الصينية واليهودية نمطية بسهولة مثل الإنجليزية والفرنسية.

عنصر التحكم والتنسيق المرئي

يضيف CSS2 المزيد من خصائص التنسيق لتوفير المزيد من التحكم الدقيق في الكائنات التي تكون المستند تستطيع الآن تحديد الأماكن والأبعاد المطلقة للعناصر يوجد أيضاً المزيد من الأنماط المعروضة لاستخدامها عند إنشاء العناصر من الممكن تطبيق الظلال في النص. كما يمكن تحديد الخطوط والألوان تماماً مثل استخدام عنصر واجهة مثل بند قائمة أو رمز لعنوان. يمكنك تغيير المؤشر الذي يظهر أمامك عندما ينتقل المؤشر فوق عناصر مختلفة.

الجداول

تسهل التطورات في خاصية العرض معالجة عناصر XML في بنية على هيئة جداول كما تسهل التحكم الأفضل في محاذاتها.

Generated Content

قد يتمكن كتاب المستندات من خلال أجهزة العد المدارة آلياً، الأنظمة المرقمة، وعلامات القوائم، من فرض تطبيقات لإنشاء معلومات سريعة، عند إعادة ترتيب المستند. من الممكن إعادة حساب الأرقام بسرعة عند أي تغيير في المسند بدلاً من مشقة إدخالها يدوياً.

أوراق النمط السمعي

لقد قامت CSS2 بدمج خصائص محددة تستطيع تغطية ميزات لنظام. تركيب المحادثة، في محاولة لجعل المعلومات المشتقة أكثر سهولة لجميع الأفراد. تلك الخصائص تمكن كاتب المستند من عنصر التحكم في التنسيق، الخطوات والخصائص الأخرى لصوت المتكلم لكل عنصر داخل المستند.

إنجازات جديدة

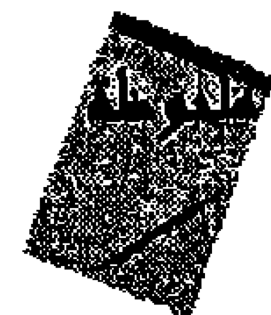
غيرت أيضاً مواصفات CSS2 تنفيذ بعض الميزات الموجودة في CSS1. وهي تحتوي على آلية التتالي الفئات المُقترحة، ومختلف الخصائِص الأخرى.

العناصر والفئات المُقترحة

يجب أن تكون الفئات المُقترحة link وvisited وactive قد صممت مستقلة عن بعضها البعض ومن الممكن استخدامها معاً.

الوراثة

قد تكون بعض الخصائص في CSS1 جاهزة لأن تأخذ قيم من الأصل. أما بالنسبة CSS2، فجميع الخصائص تستطيع أن تأخذ قيمتها من العناصر الأصل، عن طريق تعيين القيمة على لوحة المفاتيح inherit وعندما يتم أخذ خاصية، تأخذ هذه الخاصية نفس قيمة أقرب عنصر من الأصل.



بسبب إمكانية حصول كل خاصية على قيمة inherit، يتم حذف أي توضيح "شرح" لهذه القيمة في مناقشات الخصائص الفردية التالية.

آلية التتالي

يجوز للمصمم important! في CSS1 أن يقوم بفرض ورقة نمط على الكاتب لأخـــذ الأســـبقية على ورقة نمط القارئ أما CSS2 فهو يقوم بعكس تلك الأسبقية، فتأخذ تفصيلات القارئ الأسبقية على تفصيلات الكاتب. عند العمل مع الوراق النمطية لكلاً من الكاتب والقـــارئ، فـــإن النتيجـــة الافتراضية تكون تجاوز ورقة نمط المستخدم لورقة نمط الكاتب. بينما، إذا أعلن الكاتب خاصيـــة important، فإن هذا يضيف قوة أكبر للمواصفات تجعلها تتجـــاوز ورقـــة important فإنـــها تتجاوز إعلان important في ورقة نمط الكاتب بمعنى أخر يكون للقارئ الكلمة العليا.

خاصية العرض

تعد الآن القيمة الافتراضية لخاصية display وهي inline بدلاً من كونها block.

الهوامش والحواشي

لقد تم تجاهل بعض خصائص الهوامش في CSS1 عندما تم تعيين خصائص أخرى. على ســبيل المثال، يتم تجاهل margin-right إذا تم تعيين كلاً من margin-left وwidth يعتبر هذا القرار مستقلاً عن اتجاه النص ومحاذاة الكائن. ولكن CSS2 قد جعل القرار الخـــاص بتنبيـــه الـــهامش الأيمن والأيسر معتمداً على اتجاه النص والكائن.

تحديد العناصر

يستطيع المستعرضين الذين يدعموا CSS2، مثل مكتشفي الإنترنت وMozilla تحديد بدقة أكـــبر أي عنصر أو كائن يتم تطبيق خطوط النمط عليه. يمكنك باستخدام CSS2 تحديد العناصر التـــي يكون أساسها النقش الموجود بشجرة المستند عن طريق تحديد لاسم العنصر الخاص بهم، رقـــم معرف، أو من خلال خليط من العناصر وإعدادات السمات.

مطابقة النقش

يقوم مطابقة نقش CSS2 بتعريف عناصر محددة بشجرة المستند. قد يكون بناء الجملـــة لمحـــدد مطابقة النقش أي شيء من اسم العنصر البسيط إلى نظام معقد من نقوش السياق مثـــل النمـــاذج الموجودة بالجدول ١٣-١.قد يطابق أي عنصر النقش إذا كان بها جميع متطلبات النقش المحـــدد. وفي XML يحتوي هذا على حجم الأحرف. "كونها حروف لاتينية كبيرة أو صغيرة".

الجدول ١٣-١

محدد بناء الجملة الخاص بـ CSS2 لمطابقة النقش

بناء الجملة	معناها
*	يعد محدد كوني وتطابق أي عنصر
X	تطابق أي عنصر تحت اسم "X"
X Y	تطابق أي عنصر تحت "Y"، يعد من أ عد تنازلي للعنصر تحت اسم "X"، على سبيل المثال، جميع VERSE يعدان من عناصر SONNET
X > Y	تطابق أي عنصر "Y" الذي يعد تابع لأي عنصر "X". على سبيل المثال جميع توابع VERSE لعنصر STANZA
X: تابع أول	يطابق جميع عناصر "X" والتي تعد أول تابع. على سبيل المثال: أول عنصر STANZA في عنصر SONNET
X: ارتباط	يطابق جميع عناصر "X" في ارتباط هدفه لم يتم زيارته بعد
X: visited	يطابق جميع عناصر "X" الذي قد تم زيارة هدفه
X: نشط	يطابق جميع عناصر "X" التي قد تم تحديدها مؤخراً
X: hover	يطابق جميع عناصر "X" الذي يتحرك عليها مؤخراً الماوس
X: focus	يطابق جميع عناصر "X" التي حصلت مؤخراً على تركيز المستخدم إما خلال تحديدها عن طريق الماوس أو عن طريق الاستعداد لإدخال بيانات نصية
X: lang(I)	يطابق جميع عناصر "X" المصممة لاستخدام لغة الإنسان I باستخدام سمة xml:lang
X + Y	يطابق جميع العناصر "X" التي ترتبط ارتباط مباشر بعنصر "X" على سبيل المثال. عنصر REFRAIN الذي يتم سبقه على الفور بعنصر STANZA

الجدول ١٣-١

محدد بناء الجملة الخاص بـ CSS2 لمطابقة النقش

بناء الجملة	معناها
X[attr]	يطابق جميع عناصر "X" مع تعيين "attr" بصرف النظر عن قيمة السمة. على سبيل المثال: عنصر AUTHORمع سمة NAME
X[attr="string"]	يطابق جميع عناصر "X" مع يكون عنده قيمة سمة "attr" "string". على سبيل المثال: عنصر AUTHOR مع سمة DATE بقيمة ١٩٩٩٠٧٢٣
X[attr~="string"]	يطابق أي عنصر "X" الذي يكون سمته "attr" قائمة من الكلمات على مسافات متباعدة، يكون كل عنصر بها هو "string"
X[lang|="langcode"]	يطابق جميع عناصر "X" مع تعيين سمة "lang"إلى "langcode" محدد
X#myname	يطابق أي عنصر "X" يكون لسمة "id" المعرف له قيمة "myname"

رمز الاختيار الكوني

يحدد الرمز الكوني (*) جميع العناصر في المستند. يمكنك هذا من تعيين الأنماط الافتراضية لجميع العناصر على سبيل المثال، هذه القاعدة تعيين الخط الافتراضي لنيويورك.

```
* { font-face: "New York" }
```

تستطيع أن تخلط (*) مع محددي السمة، الفئة المُقترحة، والعنصر المُقترح، لتطبيق الأنماط على جميع العناصر التي لها سمة محددة، قيمة للسمة دور والخ. على سبيل المثال:

```
*:before      { content: ". " counter(pgraph) ". ";
              counter-increment: pgraph; /*Add 1 to pgraph*/
*[onmouseover] { text-decoration: blink }
```



تستطيع إلغاء (*) إذا كنت تستخدم رمز الاختيار العالمي مع مواصفة خاصية أخرى فقط.

```
before  { content: ". " counter(pgraph) ". ";
          counter-increment: para }
[onmouseover] { text-decoration: blink }
```

السلالات ومحددي التوابع

يمكنك تحديد العناصر التي تكون توابع أو سلالة لنوع محدد من عنصر لـــه محدديـــن للســـلالة والتابع. على سبيل المثال، ويمكنك تحديد أي عنصر VERSE يوجد داخل عنصـــر SONNET أو عناصر VERSE فقط التي تغير توابع مباشرة لعنصر STANZA. راجع تعليمات البرمجة ١٣-١ والتي تظهر قصيدة شكسبير رقم ٢١ في XML.

تعليمات البرمجة ١٣-١: قصيدة ٢١ لشكسبير

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="shakespeare.css"?>

<SONNET>
 <AUTHOR>William Shakespeare</AUTHOR>
 <TITLE>Sonnet 21</TITLE>
 <STANZA id="st1">
  <VERSE>So is it not with me as with that Muse</VERSE>
  <VERSE>Stirr'd by a painted beauty to his verse,</VERSE>
  <VERSE>Who heaven itself for ornament doth use</VERSE>
  <VERSE>And every fair with his fair doth rehearse;</VERSE>
 </STANZA>
 <STANZA id="st2">
  <VERSE>Making a couplement of proud compare</VERSE>
  <VERSE>With sun and moon, with earth and sea's rich
       gems,</VERSE>
  <VERSE>With April's first-born flowers, and all things
       rare</VERSE>
  <VERSE>That heaven's air in this huge rondure hems.</VERSE>
 </STANZA>
 <STANZA id="st3">
```

```
  <VERSE>O, let me, true in love, but truly write,</VERSE>
  <VERSE>And then believe me, my love is as fair</VERSE>
  <VERSE>As any mother's child, though not so bright</VERSE>
  <VERSE>As those gold candles fix'd in heaven's air.</VERSE>
 </STANZA>
 <REFRAIN>
  <VERSE>Let them say more that like of hearsay well,</VERSE>
  <VERSE>I will not praise that purpose not to sell.</VERSE>
 </REFRAIN>
</SONNET>
```

تعتبر جميع عناصر VERSE سلالة من عنصر SONNET ولكن لا يعتبروا توابع فوريــــة. تعتبر بعض عناصر VERSE توابع فورية لعناصر STANZA وبعضها الآخر يعتبر توابع فورية لعنصر REFRAIN تتكون محددي السلالات من عنصرين مصممين أو أكــثر منفصــلان عــن بعضها البعض بمسافة بعد محدد الســلالة لنمــوذج SONNET ذات VERSE مطابقــاً لعنصــر VERSE الذي يعد مستعد عشوائي عنصر SONNET. لكي تستطيع أن تحدد طبقة محــددة مــن السلالة تحتاج لاستخدام نموذج SONNET * VERSE والذي يفرض عنصــر VERSE ليكــون على الأقل تابع عنصر التابع أو مشتق منخفض لعنصر SONNET.

يجب أن تستخدم نموذج STANZA > VERSE، لكي تحدد العنصر التابع الفوري. هذا يجعل القاعدة مطبقة فقط على عناصر VERSE التي تعد تابع مباشر لعنصر STANZA ولهذا لن تؤثــر على أي توابع VERSE لعنصر REFRAIN.

يمكنك خلط كلٍ من محددي السلالة والتابع لإيجاد عناصر محددة. على سبيل المثـــال، يجــد المحدد التالي جميع عناصر VERSE التي تعد أول تابع لعنصر REFRAIN في المقــابل ســلالة لعنصر SONNET.

```
SONNET REFRAIN>VERSE:first { padding: "2cm" }
```

بالتطبيق على التعليمات البرمجية ١٣-١ تحدد هذه القاعدة للبيت الشعري "Let say them more that like of hearsay well".

محددات التابع المتجاور

يستخدم محددات التابع المتجاور علامة (+) بين مصممين العنصر ليقوموا بتعريف العنصر الذي يتلو العنصر الآخر في نفس المستوى الهيكلي. على سبيل المثال، تحدد التعليمات البرمجية التالية جميع عناصر REFRAIN التي تعمل على مشاركة الأصل مع عنصر STANZA وعلى الفــور تتبع عنصر STANZA.

```
STANZA+REFRAIN {color:red}
```

محددي السمة

يقوم محددي السمة بتعريف خليط من السمات/العناصر المحددة ضع اسم السمة التي قد تطـــابقت داخل أقواس مربعة بعد اسم العنصر. على سبيل المثال، تقوم القاعدة بتحويـــل جميـــع عنـــاصر STANZA إلى سمة NUMBER الحمراء اللون.

```
STANZA[NUMBER]  { color: red }
```

تقوم هذه القاعدة بتحويل جميع عناصر STANZA الذي له سمة NUMBER الحمراء، بغـــض النظر عن قيمة السمة. تتضمن هذه العناصر التي لها سمة NUMBER الافتراضية المزودة بـــــ DTD ولكن لا تتضمن عناصر STANZA التي ليس لها سمة NUMBER.

لكي تختبر قيم السمة، استخدم نفس بناء الجملة الذي تستخدمه لتعيين قيمة السمة في هذا يعني الاسم الذي يليه علامة يساوي، ويتلوه القيمة الموضوعة داخل علامة اقتباس. على سبيل المثـــال، لكي تحدد أن عناصر STANZA فقط التي إذا كان قيمة سمة NUMBER قيمتـــــها ٣ فيجـــب أن تتحول إلى اللون الأحمر، فيجب عليك أن تستخدم تلك القاعدة:

```
STANZA [NUMBER="3"] { color: red }
```

@r 's

اعد @ بعمل أشياء غير تحديد عنصر ما وتطبق بعض الأنماط عليه. يوجد خمسة:

- @page: تقوم بتطبيق الأنماط إلى الصفحة "معارضة للعنـــاصر التـــي توجـــد علـــى الصفحة".

٢- @import: تقوم بتضمين ورقة نمط خارجية في ورقة نمط حالية.

٣- @media: تقوم بتجميع القواعد النمطية للسمات التي يجب أن يتم تطبيقها علـــى نـــوع واحد فقط من الوسائط.

٤- @font-face: تقوم بوصف خط يستخدم في مكان أخر في ورقة النمط.

٥- @charset: تقوم بتعريف مجموعة الأحرف المستخدمة بواسطة ورقة النمط.

@page

تحدد قاعدة @page مربع الصفحة. يستطيع المصمم أن يحدد بداخلها أبعـــاد تخطيـــط، اتجـــاه وهوامش الصفحات الفردية يعد مربع الصفحة مساحة مستطيلة، تقريباً نفـــس حجـــم الصفحـــة المطبوعة التي تحتوي على مساحة الصفحة وكتلة الهامش. تحتوي مساحة الصفحة على المـــادة

التي يتم عرضها، وكما تحتوي حاشية المربع على حاوية يظهر خلالها تخطيط الصفحات بين فواصل الصفحات خلاف المربعات الأخرى، لا يوجد بمربعات الصفحة أي حدود أو صفحات، يوجد فقط إلى هوامش.

تحدد قاعدة page@ كل صفحة في المستند. يمكنك استخدام إحدى خصائص الفئة المُقترحة للصفحة: first, :left أو right: لتحدد الخصائص المختلفة لمختلف الفئات من الصفحة.

عدم إدراك قاعدة page@ بمحتوى الصفحة التي تتضمن خطوطها، فهي لا تستطيع فهم المقياس ems و s ex تعد جميع وحدات القياس الأخرى مقبولة، بما في ذلك النسب المئوية. تعد النسب المئوية المستخدمة في إعدادات الهوامش، نسب مئوية لإجمالي مربع الصفحة. من الممكن أن يكون للهوامش قيم سلبية، تضع المحتوى خارج المساحة التي تقوم بالتشغيل المعهود بواسطة التطبيق أو الطابعة في معظم الحالات، تختصر المعلومة بسهولة إلى المساحة المرئية أو المطبوعة.

@import

تعمل قاعدة import@ على تضمين ورقة نمط خارجية محددة في ورقة من أجزاء صغيرة وسهلة الفهم. تستخدم أوراق النمط المستوردة ملحق css. على سبيل المثال، تستورد القاعدة التالية ملف css للشعر.

```
@import url(poetry.css);
```

كما يمكن لقاعدة import@ تحديد نوع الوسائط التي تلي اسم ورقة النمط. فإذا لم يتم تحديد أن نوع للوسائط، فإن قاعدة import@ تكون غير شرطية، وسيتم استخدامها مع كل أنواع الوسائط. على سبيل المثال، تستطيع القاعدة التالية استيراد ملف css للوسائط المطبوعة. وسيتم تطبيق التعريفات في ورقة النمط هذه على الوسائط المطبوعة.

```
@import url(printmedia.css) print;
```

تقوم القاعدة التالية باستيراد ملف css المتواصل الذي سيتم استخدامه لكلاً من شاشات الكمبيوتر و/أو شاشة:

```
@import url(continuous.css) tv, screen;
```

تكون للأوراق النمطية التي تم استيرادها من بين أوراق نمطية أخرى في رتبة مخصصة في التتالي أقل من ورقة النمط المستوردة على سبيل المثال، لنفرض أن قام شكسبير css بعمل VERSE نمطي في خط نيويورك، بينما قام shakeprint.css بتنميط VERSE في خط Times. إذا قام شكسبير css باستيراد shakeprint.css ، فإن بيوت الشعر سيتم جعلها نمطية في نيويورك بينما، إذا قام shakeprint.css باستيراد شكسبير css، فإن بيوت الشعر سيتم تنميطها في Times.

@media

تستخدم أنواع كثيرة من الوسائط في نقل المعلومات للقارئ ويكون لكل نوع من الوسائط أنماطـها وتنسيقاتها المخصصة. ولا يمكنك أن تحصل على متناول للمحادثة يقوم بقراءة شكسـبير بنغمـة أحادية فهل يمكنك الآن؟ ولا يكون للنمط المائل شكل جيد مقبول على طرفية أحادية المسافة.

تمكنك CSS2 من تحديد أنماط مختلفة للعنصر الواحد المعروض في الوسائط على الشاشة إذا تم استخدام خط ليس رقعة، بينما تعد قراءة النص المكتوب على ورقة عامة أكثر سهولة إذا كـان مكتوب بخط الرقعة. تستطيع أن ترفق القواعد النمطية الموجهة لمتوسط واحد فقط فـي قـاعدة @media التي تعرف تلك الوسائط. من الممكن أن يوجد العديد مـن قواعـد @media فـي المستند مثلما يوجد أنواع الوسائط الذي يتم تحديدها. على سبيل المثال، تقوم تلك القواعد بتنسـيق SONNET مختلفة تعتمد على إذا ما كانت مطبوعة على ورق أو معروضة على الشاشة.

```
@media print {
  SONNET { font-size: 10pt; font-family: Times, serif }
}
@media screen {
  SONNET { font-size: 12pt;
        font-family: New York, Times New Roman, serif }
}
@media screen, print {
  VERSE { line-height: 1.2 }
}
```

تقوم أول قاعدتين بتعريف أنماط محددة لأنواع الوسائط المرئية على الشاشة والمطبوعة على التوالي وبما أن شاشات الكمبيوتر الحديثة أقل دقة من الطابعات الحديثة، فمن المهم جعل الخـط على الشاشة أكبر من على النسخة المطبوعة، ومن المهم اختيار خط مصمم لاسـتخدامه علـى الشاشة.

توفر القاعدة الثالثة المزيد من الأنماط التي يتم تطبيقها على كل نوع من أنواع هذه الوسـائط. ولكي تصمم إرشادات نمطية لأنواع الوسائط المتعددة فورياً، تستطيع بسهولة سردهم بعد مصمـم قاعدة @media وتفصلهم بفاصلة.

يستطيع المستعرضين الذين يدعموا CSS2 السماح لكاتب المستند بتوفير قواعد تنص علـى كيفية عرض مستند ما لنوع معين من الوسائط. على سبيل المثال، ستقوم بتطبيق القواعد المختلفة عند إظهار مستند على الشاشة بطريقة أفضل من إرساله إلى الطابعة. يعرف CSS2 عشرة أنواع للوسائط وهما:

١ – all: جميع الأجهزة.

٢ – aural "متواصل/سمعي": متناول للمحاولة.

٣ – Braille "المتواصل، الملموس": أجهزة braille الملموسة لاسترجاع الشكل التالف.

٤ – Embossed "مرقمة، ملموسة": طابعات braille المرقمة.

٥ – Handheld "مرئي": أجهزة PDAs والأجهزة الأخرى المحمولة يدوياً مثـــل ويـنـدوز palmtops وnewtons وPalm Pilots.

٦ – Print "مرقمة، مرئياً": جميع المواد المطبوعة، وغير شفافة.

٧ – Projection "مرقمة، مرئياً": العروض التقديمية وعروض الشرائح سواء عن طريـــق إبرازها مباشرة من الكمبيوتر أو عن طريق طباعتها على أسطح شفافة.

٨ – Screen "متواصل، مرئي": شاشات الكمبيوتر الملونة، ذات الصور النقطية.

٩ – Tty "متواصل، مرئي": المحطات الطرفية الصامتة وشاشات أجهزة الكمبيوتر القديمـــة التي تستخدم خطوات ثابتة وشبكة للأحرف أحادية الكروم.

١٠ – Tv "سمعي، مرئي": أجهزة الكتابة التلفزيونية. على سبيل المثال، ذات الدقة المنخفضـة، شاشة متشابهة ملونة.

لا يحتاج برنامج الاستعراض إلى تدعيم كل هذه الأنواع. في الواقع، أنا لا أعرف أي جـــهاز فردي يقوم بتدعيم كل هذه الأنواع، أنا لا أعرف أي جهاز فردي يقوم بتدعيم كل هـــذه الأنـــواع بينما يجب على مصممي ورق النمط افتراض أن القراء سيقومون باستخدام كل هذه الأنواع مـــن الأجهزة ليروا محتواها.

بالطبع، تتغير مع مرور الوقت ميزات الوسائط الفردية لقد كانت آلة الطباعة الأولى الخاصـة بي ١٤٤ نقطة فـــي البوصـــة ولكـــن مثـــل هـــذه الطباعـــات ذات الدقـــة المنخفضـــة يجـــب أن تكون نسبياً نادرة في القرن الواحد والعشرون. وعلى الجهة الأخرى، فإن الشاشات ســـتصل إلى دقة بنسبة ٣٠٠٠ نقطة في البوصة أو أكثر، وستتوافر الطباعة الملونة سريعاً للمزيـــد مـــن المستخدمين.

من الممكن أن تتواجد بعض الخصائص فقط في أنواع محددة من الوسائط. على سبيل المثال، فإن خاصية pitch تكون منطقية فقط مع نوع الوسائط السمعية. ولا يقوم CSS2 بتحديـــد قائمـــة شاملة لأنواع الوسائط، على الرغم من أنها توفر قائمة من القيم الحالية لقاعدة @media. ولـــم تعد تلك القيم حجم الأحرف.

@Font-face

توفر قاعدة font-face@ لنا مواصفات المحرف المستخدم في أي مكان أخر في ورقة النمط يمكنها أن تمدنا باسم الخط، URL الذي يمكن من خلاله تحميل الخط، ومعلومات مفصلة عن القياس المتري للخط الذي يسمح بتركيب "تبادل" فاكس معقول. كما تتحكم قاعدة font-face@ أيضاً في الكيفية التي يحدد بها البرنامج الخطوط لمستند له خطوط محددة من الكاتب يمكنك أن تفترض تطابق خطي متماثل، تطابق خطي ذكي تركب الخط المطلوب، تقوم بتحميل الخطوط من الخادم، أو معالجة الخط. تلك الأساليب موصوفة فيما يلي:

- **تطابق خطي متماثل:** يقوم برنامج المستخدم باختيار خط النظام المحلي مع نفس اسم الأسرة. وليس من الضروري أن تطابق الخطوط التي لها نفس الاسم في المظهر. من الممكن أن يكون الخط الذي يستخدمه العميل من مصدر مختلف عن الخط الموجود على الخادم.
- **تطابق خطي ذكي:** يقوم البرنامج باختيار خط متوافر في نظام العميل، وقريب جداً في المظهر من الخط المطلوب. ولا يعد هذا التطابق بالضبط، ولكن يجب أن يكون قريب منه. يعتمد أساساً الخط المتطابق على نوع الخط، عن ما إذا كان يستخدم الرقعة، حجمه، ارتفاع الأحرف الأولى له، وميزات أخرى للخط.
- **تركيب الخط:** يقوم استعراض ويب ببناء خط تقريباً يشبه الخط المصمم، ويشاركه في مقياسه المتري. عندما يتم تركيب الخط فسيكون عامةً تكرار قريب جداً له عن كونه خط يتم إيجاده بالمطابقة. يتطلب التركيب استبدال دقيق ووضع المعلومات بطريقة مرئية لكي يتم المحافظة على جميع ميزات الخط.
- **تحميل الخط:** يقوم برنامج الاستعراض بتحميل الخط من URL محدد تعد هذه العملية نفس عملية تحميل أي صورة أو صوت ليتم عرضهم مع المستند الحالي. وسيتوقع بعض المستخدمين الذين يقومون بتحميل الخطوط بعض التأخيرات المشابهة للتأخيرات التي تظهر عند تحميل صورة.
- **معالجة الخط:** يعد أخر بديل لإدارة الخطوط هو الأداء المتقدم. يعتبر هذا خليط من التحميل والتطابق الذي يمكن المستعرض من إنشاء خط مؤقت ليتم قراءة محتوى المستند، بينما يتم تحميل الخط الأصلي. بعد تحميل الخط الحقيقي، فإن هذا الخط الحقيقي يحل محل الخط المركب في المستندات المتتالية ولكي تتجنب معالجة المستند مرتين، يجب أن تحتوي مواصفات خطك على المعلومات المترية التي تصف الخط كلما كانت المعلومات المترية للخط كاملة، كلما قل احتياج المستند لإعادة معالجته بمجرد أن يكون التحميل كامل.

يمكن CSS2 كاتب المستند من تحديد أياً من تلك الأساليب إذا أمكن، استخدامها عندما لا يتم توافر أي خط مصمم مواصفات الخط، الذي تم إنشاء من بين سلسلة من الذين يقومون بوصف الخط، وتقوم بتعريف المعلومات المفصلة عن الخطوط التي تستخدم على الصفحة. ويتميز كل واصف للخط بجزء من المعلومات محددة عن الخط. ومن الممكن أن تتضمن تلك المواصفات URL للخط، الاسم العائلي للخط، وحجم الخط.

ينقسم من يقومون بوصف الخط إلى ثلاثة أنواع:

- الذين يقومون بتوفير رابطة بين استخدام ورقة نمط الخط ومواصفاته.
- الذين يقومون بتوفير URL لمكان الخط أو المعلومات وثيقة الصلة به.
- الذين يقومون بتوفير معلومات حرفية عن الخط.

تطبق قاعدة @font-face فقط على الخطوط المحددة داخل ورقة النمط. سوف تحتاج إلى إحدى مواصفات قاعدة @font-face لكل خط في ورقة النمط على سبيل المثال:

```
@font-face { font-family: "Comic Sans";
        src: url(http://metalab.unc.edu/xml/fonts/comicsans)}
@font-face { font-family: "Jester"; font-weight: bold;
            font-style: italic}
TITLE      { font-family: "Comic Sans"}
AUTHOR     { font-family: "Jester", serif}
```

بما أن البرنامج يقوم بقراءة ورقة النمط، فستتناول إيجاد مجموعة من القواعد التي تحدد كيف يتم معالجة كل عنصر تقوم ورقة النمط بتعيين جميع العناصر الخاصة TITLE لأصل الخط Comic Sans وفي الوقت نفسه تعيين جميع العناصر الخاصة AUTHOR لخط Jester وسيقوم تطبيق استعراض ويب الذي يدعم CSS1 بالبحث عن خط منصوص عليه لأصول خط Comic Sans وJester font وإذا لم تحدهم، فستستخدم الخط المنصوص عليه الافتراضي لأصل Comic، خط الرقعة المحدد المائل للخلف. سيتم تجاهل واصفي خط قاعدة @font-face. وسيكون برنامج CSS1 مستعداً لتخطي هذا الأمر دون احتساب أي خطأ.

وستقوم تطبيقات التي تدعم CSS2 باختبار قواعد واجهة @font-face في محاولة لتطابق مواصفات الخط بخطوط Comic Sans وjester. في المثال السابق، سيجد البرنامج المستعرض URL الذي يمكن عن طريقها تحميل خط Comic Sans إذا تم إيجاد خط Comic Sans في نظام العميل كان البرنامج سيستخدمه بدلاً من تحميل الخط وفي حالة Jester، يستخدم مستخدمي البرنامج إحدى قواعد المطابقة، أو قاعدة التركيب لإنشاء خط مشابه من الصفات المتوافرة. إذا كان استعراض ويب لا يستطيع إيجاد مطابقة قاعدة @font-face لعائلة الخط المحددة، كان سيحاول مطابقة الخطوط التي تستخدم القواعد الخاصة بـ CSS2.

كما يمكن CSS2 أي واصف للخط غير معروف، أو مفيد للمستعرض أن يتخطاه. مما يوفر وسائل للبناء الداخلي لزيادة الواصفين، في محاولة جاهدة لتحسين القواعد المستخدمة مثل استبدال الخط، مطابقته أو تركيبه.

@charset

يوجد هناك ثلاث طرق لتحديد مجموعة الأحرف التي يتم استخدامها في كتابة ورقة النمط، وتأخذ تلك الطرق الأسبقية بالترتيب التالي:

١- معامل Charset HTTP في مجال "نوع-المحتوى"

٢- قاعدة @charset.

٣- السمات والخصائص المرتبطة بالمستند، مثل سمة HTML's Charset المستخدمة مع عنصر LINK.

من الممكن أن تحتوي كل ورقة نمط على قاعدة @charset الفردية ويجب أن تظهر قاعدة @charset في مستهل المستند، ولا يجوز مسبقاً بأي أخرف أخرى. أن بناء الجملة المستخدمة @charset هي:

```
@charset "character set name"
```

يجب أن يكون اسم مجموعة الأحرف المحددة في تلك العبارة اسم كما هو موصوف في مسجل IANA. تستطيع أن ترى قائمة جزئية لمجموعة أحرف في الجدول ٧-٧ في الفصل السابع ولكي تحدد أن ورقة النمط مكتوبة باللاتينية -١ فستكتب:

```
@charset "ISO-8859-1"
```



يتم مناقشة مجموعات الأحرف بالتفصيل في الفصل السابع، اللغات الأجنبية والنص الغير روماني.

العناصر المُقترحة

يتم معاملة العناصر المُقترحة كعناصر في أوراق النمط ولكن ليس من الضرورة أن تكون عناصر محددة في مستند XML. عبارة عن مستخلصات من أجزاء معينة من المستند المعالج بعد تطبيق ورقة النمط، على سبيل المثال، أول خط من الفقرة لا يوجد العناصر المُقترحة حجم أحرف، ومن الممكن أن تظهر فقط مباشرة بعد موضوع محدد ورقة النمط. يقوم CSS2 بتحديد عنصرين مُقترحين جديدين after و before.

تقوم العناصر المُقترحة after وbefore بتحديد المكان على الفور قبل وبعد العنصر الـــذي يسبقهم. تستخدم خاصة المحتوى في وضع بيانات في تلك المكان على سبيل المثال، تضع القاعدة السلسلة ---------- بين كائنات STANZA لمساعدتها في فصل المقاطع الشــعرية توضـــع التعليمات الجمركية فواصل الأسطر على شكل ١٨ في سلسلة حرفية.

STANZA:after {content: "\A-----\A"}

تستطيع استخدام واحد من الكلمات الأساسية الأربعة، بالإضافة إلى السلسلة الحرفيـــة، فـــإن خاصية content تعد:

١- open-quote

٢- close-quote

٣- no-open-quote

٤- no-close-quote

يعمل كلاً من الكلمــــات الأساســية open-quote وclose-quote علـــى إدراج الحـــرف الموضوع عليه علامة اقتباس مناسبة للغة والخط الحاليين (على سبيل المثال، "or") كما لا يعمــل كلاً من الكلمات الأساسية no-open-quote وno-close-quote على إدراج أي أحرف ولكـــن يعملان على تزايد مستوى التضمين وكأنه تم استخدام علامة الاقتباس. فمع كـــل مســـتوى مـــن التضمين، فإن علامات الاقتباس تتحول من الضعف للفردي أو على العكس.

ومن الممكن أيضاً أن تستخدم الدالة attr(X) مثل قيمة خاصية المحتوى لتقوم بإدراج قيمـــة سمة X قبل أو بعد العنصر المعرف.

أخيراًن تستطيع إدراج القيمة الحالية لجهاز العد الآلي باســـتخدام إمـــا دالـــة جـــهاز العـــد counter() أو counters() ويكـــون لـــــها شـــكلين مختلفيـــــن: counter(name) أو counter(name, style) يعد النمط الافتراضي عدد عشري.

الفئات المُقترحة

يحدد محددي الفئة المُقترحة العناصر التي يكون أساسها الأوجه وليس الاسم، للسه [illegible] ـتـــوى العنصر على سبيل المثال، قد تعتمد الفئة المُقترحة على موقع الماوس، الكائن [illegible] ـم التركـ [illegible] عليه، أو إذا كان الكائن رابطة أم لا. وقد يغير العنصر تكراراً [illegible] ـمُقترحة الخاصـ
عندما يكون القارئ متفاعل مع المستند تكون بعض الفئات [illegible]ـرحة خاصــــة بالتبــادل. ولكـــن معظمها يمكن تطبيقه على الفور لنفس العنصر، ويمكن وضعها في أي مكان داخل محدد العنصر عندما تقوم الفئات المُقترحة بأي تعارض فإن الترتيب المتتالي يحدد أي قاعدة يتم تنشيطها.

التابع الأول

يقوم التابع الأول للفئة المُقترحة بتحديد CSS1 :link للعنصر المسمى بغض النظر عن نوعـــــه. على سبيل المثال، في التعليمات البرمجية ١٣-١، إن عنصر CSS2, :link الذي يكون محتـــواه "هل هي ليست معي كما هي مع ذلك المتأمل" سيكون التابع الأول لعنصر AUTHOR وقد يتـــم تصميمها عن طريق تلك القاعدة.

```
STANZA:first-child {font-style: bold}
```

:link, :visited, :active

في CSS2, :link يكون للفئات المُقترحة :link, :visited, :active تبـــادل خـــاص. أمـــا فـــي CSS2, :link فإن :link, :visited لا يتركوا في تبادل خاص "لأنهم منطقياً يجـــب أن يكونـــوا كذلك"، ولكنك تستطيع أن تستخدم واحدة من تلك في اتحاد active على سبيل المثال، فإن جـــزء التغييرات البرمجية التالية يفترض أن قد تم عنصر AUTHOR مثل الربط ويحول ألوان النـــص معتمداً على حالة الربط الحالية. وفي جزء التغييرات البرمجية التالية، بتعيين الربط الذي لم يتـــم زيارته باللون الرمادي، أما الربط النشط فيتم ظهوره باللون الأخضر الجيري بينما يكون المؤشر فوقه.

```
AUTHOR:link    { color: "red" }
AUTHOR:visited { color: "gray" }
AUTHOR:active  { color: "lime" }
```

:hover

يحدد الفئة المُقترحة :hover العناصر يشير إليها الماوس أو أجهزة الإشارة الأخرى، ولكن بدون ضغط زر الماوس. على سبيل المثال، إن هذه القاعدة تقوم بتلوين عنصــــر AUTHOR بـــاللون الأحمر عندما يشير إليه المؤشر:

```
AUTHOR:hover { color: "red" }
```

يعوم عنصر AUTHOR إلى لونه الطبيعي عندما يتوقف المؤشر من الإشارة إليه

:focus

تشير الفئة المُقترحة focus إلى العنصر الذي يكون عليه التركيز حالياً. يكـــون التركـــيز علـــى العنصر عندما يتم تحديده ويكون جاهزاً لاستقبال بعض أنواع إدخال النص. تجعل القاعدة التاليـــة التركيز على العنصر بخط سميك.

```
:focus { text-style: "bold" }
```

:lang()

تحدد الفئة المُقترحة ()lang العناصر التي لها لغة محددة. ويتم ذلك عاماً في XML بواسطة سمة xml:lang و/أو سمة وضع التغييرات البرمجية لإعلان تعريف XML تقوم القاعدة التالية بتغيير اتجاه جميع عناصر VERSE المكتوبة باللغة اليهودية ليتم قراءتها من اليمين إلى الشمال، بدلاً من الشمال إلى اليمين:

```
VERSE:lang(he) {direction: "rtl" }
```

:right, :left, :first

تعد الفئات المُقترحة right وleft: وfirst: مطبقة فقط على قاعدة page rule@ فهي تمكنك من تحديد الأنماط المختلفة للصفحة الأولى من المستند، وللصفحات اليسرى للمستند "عامة مرقمة زوجي" وللصفحات اليمنى للمستند "عامة مرقمة فردي". على سبيل المثال، تقوم تلك تقوم القواعد بتحديد الهوامش الكبيرة جداً.

```
@page:right  { margin-top: 5cm;
               margin-bottom: 5cm;
               margin-left: 7cm;
               margin-right: 5cm }
@page:left   { margin-top: 5cm;
               margin-bottom: 5cm;
               margin-left: 5cm;
               margin-right: 7cm }
@page:first  { margin-top: 10cm;
               margin-bottom: 10cm;
               margin-left: 10cm;
               margin-right: 10cm }
```

الوحيدة التي يمكنك تعيينها في القاعدة لتلك الفئات المُقترحة هي الخصائص الخاصة بالهامش.

تنسيق صفحة

يقوم محدد page@ بالإشارة إلى صفحة وهو معتاد على تعيين الخصائص المطبقة على الصفحة نفسها بدلاً من عنصر XML الفردي على الصفحة. يوجد لكل صفحة في المستند مجموعة متنوعة من الخصائص التي تطبق عليها، بما في ذلك حجم الصفحة، الاتجاه، الهوامش، وفواصل الصفحات. تتلو هذه الخصائص أي عنصر موجود على الصفحة. ويمكن للفئات المُقترحة الاختيارية أن تحدد خصائص مختلفة للصفحة الأولى، الصفحات ذات الأوجه اليمنى والصفحات ذات الأوجه اليسرى

يقوم CSS2 بالافتراض المعقول أن تكون الصفحات مستطيلة. بتوافر هذا الافتراض يجـــوز للصفحة أن تضع خصائص المربع المألوفة لك في نظام CSS1 بما في ذلك الهوامش والحجـــم. بينما، لا يوجد لمربع الصفحة أي حدود أو مسافة فارغة بينها وبين البيانات، لأنهم من الطبيعـــي أن يكونوا موجودين خارج الصفحة الحقيقية.

خاصية الحجم

في قاعدة @page تحدد خاصية size ارتفاع وعرض الصفحة. يمكنك تعييـــن size بواحـــد أو اثنين من الأطوال المطلقة أو كواحـــد مـــن الكلمـــات الأساســـية الأربعـــة auto, portrait, landscape وinherit فإذا تم إعطاء طوال واحد فقط فإن الصفحة ســـتكون مربعـــة. فـــإذا تم إعطاء الأطوال كلها، فإن الأولى هي عرض الصفحة، والثانية هي طولها على سبيل المثال،

```
@page { size: 8.5in 11in }
```

تأخذ إعدادات auto حجمها آلياً على الورقة أو الشاشة الهدف. أما landscape فهي تفرض على المستند أن يكون منسق لتناسب الصفحة الهدف، ولكن بجوانب أفقيـــاً طويلـــة كمـــا تنســـق إعدادات portrait المستند ليناسب حجم الصفحة ذات الهدف الافتراضي، ولكن بجوانب رأســـية طويلة.

خاصية الهامش

تتحكم خاصية margin في هوامش الصفحة ---- المساحات المستطيلة الموجودة في الجوانـــب الأربعة التي لا يكون مطبوع عليها أي شيء تستخدم هذه الخاصية كاختزال لتعييـــن خصـــائص margin-top وmargin-bottom وmargin-right وmargin-left كلاً على حدة تعـــد هـــذه الخصائص نفس الخصائص الخاصة بالمربعات في CSS1 على سبيل المثال، تصف هذه القـــاعدة ٨,٥ في ١١ بوصة صفحة مع بوصة واحد من الهوامش في جميع الجوانب.

```
@page { size: 8.5in 11in; margin: 1.0in }
```

خاصية الترميز

يوفر لنا CSS2 لتضع العلامات على صفحة ترسم أين تتم قطع الأوراق و/أو كيف يتم محـــاذاة الصفحات. تظهر تلك العلامات خارج مربع للصفحة يصبح مربع الصفحة بسهولة أكثر مســـاحة مرئية في المستند الذي قد يتأثر بقاعدة @page إذا كنت تنظر إلى جزء من الورقة مطبـــوع 8 1/2 "x11" سيكون مربع الصفحة كل شيء داخل المنطقة المطبوعة على تلك الورقة، وهو مـــا نعتقد طبيعياً أنه المسافة داخل الهوامش المطبوعة. يتحكم البرنامج في أداء العلامات، التي يتـــم

عرضها فقط على مربعات للصفحة مطلقة لا تستطيع المربعات المطلقة للصفحـــة أن تتحـرك، ويمكن التحكم فيها عن طريق الهوامش العامة للصفحة. يتم محاذاة المربعات النسبية للصفحة ضد الصفحة الهدف، في معظم الحالات التي تفرض عدم وضع علامات على طرف الصفحة. عنـــد محاذاة مربع نسبي للصفحة التي توجد في عقلك، وتستخدم خصائص margin و padding فـــي تحريك المساحة المطبوعة على تلك الصفحة عن الورقة الحقيقية.

يوجد لخاصية mark أربعة قيم ممكنة crop و cross و inherit و none ويمكن اســـتخدامهم فقط مع عنصر page@ تقوم علامات Crop بتعريف الأطراف المقطوعة من الورقـــة تعـرف أيضاً علامات Cross بعلامات التسجيل، وتستخدم لمحاذاة الصفح بعد الطباعة. إذا تعينت القيمـة none، فلا يتم عرض أي علامات على المستند. تحدد القاعدة التالية صفحة بها كلاً من علامات crop و cross.

```
@page { mark: crop cross}
```

خاصية الصفحة

بالإضافة إلى استخدام محدد page@ لتحديد خصائص الصفحة، يمكنك ربط خصائص الصفحـة بالعناصر الفردية التي تستخدم خاصية page@ ولتقوم بعمل ذلك تكتب قـــاعدة page@ التـــي تحدد خصائص الصفحة، وتعطي لقاعدة page@ تلك اسم وبعدها تستخدم الاســـم مثـــل قيمـة خاصية الصفحة لقاعدة عنصر عادي على سبيل المثـــال، تقـــول هاتـــان الصفحتـــان معـــاً أن SONNET سيتم طبعها في اتجاه أفقي.

```
@page rotated { size: landscape}
SONNET       { page: rotated}
```

عند استخدام خاصية page من الممكن العناصر المتقاربة المختلفـــة أن تحـــدد الخصـــائص المختلفة للصفحة فإذا حدث ذلك، سيتم إدراج فاصل للصفحة بين العناصر. إذا اســـتخدم التـــابع الأسبقية. على سبيل المثال، في المثال التالي يتم تقديم الجدولين على صفحات أفقية، ومن الممكن أن يكون على نفس الصفحة إذا كانت هناك مساحة. بسبب وضع العناصر على شكل طبقات فـــي المستند، فإن تعيين الصفحة الدائرية لعنصر SONNET يتم تجاوزه، ولا يستخدم.

```
@page narrow  { size: 9cm 18cm}
@page rotated { size: landscape}
STANZA       { page: narrow}
SONNET       { page: rotated}
```

خصائص فاصل الصفحة

تقوم خاصية page-break-after بفرض أو منع إدراج فاصل الصفحة بعد الكائن الحالي. أمـــا خاصية page break before فتقوم بفرض أو منع إدراج فاصل الصفحة قبل الكائن الحـــالي.

كما تقوم خاصية page break inside بالسماح أو المنع لإدراج فاصل للصفحة داخل الكائن الحالي. من الممكن استخدام تلك الخصائص معاً للحفاظ على تماسك الفقرات التي لها نص عناوين مترابطة ولها نص هيكلي وصور وعناوين مترابطة، وللمحافظة على الجداول المتكاملة في نفس الصفحة.

عندما يتم تعيين أياً من تلك الخصائص عند auto لا يتم فرض أو منع فاصل الصفحة بعد المربع الحالي وتفرض. إعدادات always فاصل للصفحة. تمنع إعدادات avoid، ظهور فاصل الصفحة. كما تفرض الإعدادات left وright إدراج واحد أو اثنين من فواصل الصفحة بالضرورة لكي تفرض على الصفحة التالية أن تكون إما صفحة على شرحها left أو right. ويعد هذا مفيداً في نهاية أي فصل في أي كتاب تبدأ فيه الفصول عامة بصفحات اليد اليمنى حتى إذا تركت صفحات فارغة.

تعمل القاعدة التالية على إدراج فاصل للصفحة قبل وبعد كل عنصر SONNET في المستند، ولكن ليس بداخل القصيدة حتى تظهر كل قصيدة في صحفتها الخاصة بها.

```
SONNET { page-break-before: always;
         page-break-after:  always;
         page-break-inside: avoid }
```

التنسيق المرئي

يضيف CSS2 العديد من ميزات التنسيق الجديدة التي توفر تحكم أكبر على تخطيط مستند XML الخاص بك. ولخاصية display العديد من القيم الجديدة التي تمتد على الكتلة الرئيسية والأنواع المحولة للـ CSS1. تمكنك خاصية المؤشر من التعريف على أي نوع من المؤشرات تعرضه على الكائن الخاص بك. تستطيع التحكم في ارتفاع وعرض جميع مربعات الكائن. كما تعطيك أيضاً CSS2 القدرة على تعديل الوضوح، الحجم الملتقط، اللون، الخط، ظلال النصر، والمحاذاة الخاصة بكائن مستندك كما تتحكم في كيفية التعامل مع محتويات الكائن إذا تطلب ظهور أي تجاوز للسعة.

خاصية العرض

يمنحك التوسع في خاصية display في CSS2 المزيد من خيارات التخطيط الكاملة، وبالأخص الجداول. في CSS2، هناك سبعة عشر قيمة ممكنة لخاصية العرض:

Inline	table-header-group
Block	table-footer-group

الجدول ١٣-١

محدد بناء الجملة الخاص بـ CSS2 لمطابقة النقش

بناء الجملة	معناها
X[attr]	يطابق جميع عناصر "X" مع تعيين "attr" بصرف النظر عن قيمة السمة. على سبيل المثال: عنصر AUTHORمع سمة NAME
X[attr="string"]	يطابق جميع عناصر "X" مع يكون عنده قيمة سمة "attr" "string". على سبيل المثال: عنصر AUTHOR مع سمة DATE بقيمة ١٩٩٩٠٧٢٣
X[attr~="string"]	يطابق أي عنصر "X" الذي يكون سمته "attr" قائمة من الكلمات على مسافات متباعدة، يكون كل عنصر بها هو "string"
X[lang|="langcode"]	يطابق جميع عناصر "X" مع تعيين سمة "lang"إلى "langcode" محدد
X#myname	يطابق أي عنصر "X" يكون لسمة "id" المعرف له قيمة "myname"

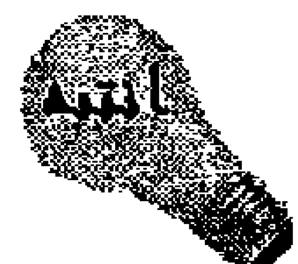
رمز الاختيار الكوني

يحدد الرمز الكوني (*) جميع العناصر في المستند. يمكنك هذا من تعيين الأنماط الافتراضية لجميع العناصر على سبيل المثال، هذه القاعدة تعيين الخط الافتراضي لنيويورك.

```
* { font-face: "New York" }
```

تستطيع أن تخلط (*) مع محددي السمة، الفئة المُقترحة، والعنصر المُقترح، لتطبيق الأنماط على جميع العناصر التي لها سمة محددة، قيمة للسمة دور والخ. على سبيل المثال:

```
*:before        { content: ". " counter(pgraph) ". ";
                  counter-increment: pgraph; /*Add 1 to pgraph*/
*[onmouseover] { text-decoration: blink }
```



تستطيع إلغاء (*) إذا كنت تستخدم رمز الاختيار العالمي مع مواصفة خاصية أخرى فقط.

```
before  { content: ". " counter(pgraph) ". ";
         counter-increment: para }
 [onmouseover] { text-decoration: blink }
```

السلالات ومحددي التوابع

يمكنك تحديد العناصر التي تكون توابع أو سلالة لنوع محدد من عنصر لــه محدديــن للســلالة والتابع. على سبيل المثال، ويمكنك تحديد أي عنصر VERSE يوجد داخل عنصــر SONNET أو عناصر VERSE فقط التي تغير توابع مباشرة لعنصر STANZA. راجع تعليمات البرمجة ١٣-١ والتي تظهر قصيدة شكسبير رقم ٢١ في XML.

تعليمات البرمجة ١٣-١: قصيدة ٢١ لشكسبير

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="shakespeare.css"?>

<SONNET>
  <AUTHOR>William Shakespeare</AUTHOR>
  <TITLE>Sonnet 21</TITLE>
  <STANZA id="st1">
    <VERSE>So is it not with me as with that Muse</VERSE>
    <VERSE>Stirr'd by a painted beauty to his verse,</VERSE>
    <VERSE>Who heaven itself for ornament doth use</VERSE>
    <VERSE>And every fair with his fair doth rehearse;</VERSE>
  </STANZA>
  <STANZA id="st2">
    <VERSE>Making a couplement of proud compare</VERSE>
    <VERSE>With sun and moon, with earth and sea's rich
        gems,</VERSE>
    <VERSE>With April's first-born flowers, and all things
        rare</VERSE>
    <VERSE>That heaven's air in this huge rondure hems.</VERSE>
  </STANZA>
  <STANZA id="st3">
```

أما إذا كان المربع الذي يليه ليس مربع كتلة، أو كان مربع الضغط لا يناسب الهامش، فــهي تعامل ببساطة كأنها مربع كتلة أخر

تمكنك قيمة run-in من تنسيق عناصر متكتلة عادية لتكون الكتلة المحولة الأولــى لعنصــر الكتلة التالي في التعليمات البرمجية. إذا كان العنصر التالي ليس عنصر كتلة، إذن ينسق عنصــر السطر نفسه كعنصر كتلة.

قيمة العلامة

يقوم إعداد خاصية display وmarker بتعريف كتلة تكونت بواسطة المحتوى التي نتجت مــن ورقة القيمة فقط مع العناصر المُقترحة before وafter التي قد ارتبطت بعناصر مستوى الكتلة.

قيم عرض الجدول

تعد واحدة من أهم الميزات الجديدة في CSS2، خاصة لمبرمجي XML الذين ينشئون دائماً بنيــة حدودية بعلامات لا تشبه إطلاقاً علامات جدول XML، وهو دعم لتخطيط العناصر بالجدول. كما تضيف CSS2 الدعم للعناصر النمطية، فهناك أجزاء من الجداول تستخدم تلــك القيــم العشــرة لخاصية العرض:

١-Table	٢-جدول – صف
٣-inline-table	٤-جدول – عمود – مجموعة
٥-table-row-group	٦-جدول – عمود
٧-table-header-group	٨-جدول – خلية
٩-table-footer-group	١٠- جدول – عنوان

على سبيل المثال، إن إعداد خاصية display إلى table تشير إلى أن العنصر المحدد يعــد محتوى مستوى. الكتلة لمختلف التوابع الصغيرة التي سيتم ترتيبها في شبكة تفرض قيمة inline-table على الجدول أن يكون له وظيفة الكائن المحول تمكن النص لينتقل على جاذبيه، ويمكــن الجداول المتعددة أن توضع جنباً إلى جنب. تنسق قيمة table caption عنصر ليكــون جــدول معنون كما تنشئ قيم كلاً مــن table-row-group وtable-header-group و table-footer-grou مجموعات من خلايا البيانات التي تعمل كصف فردي، كأنها قد تم تعريفها باستخدام قيمــة table-row تنشئ table-column-group مجموعة من خلايا البيانات التي تعمل كعمود فردي الذي تعريفه باستخدام قيمة table-column ولعناصر XML التي تظهر في كفايتها الطبيعية مــن خاصية display بقيمة table-cell خلايا الجدول.

على سبيل المثال، إذا كنت تقوم بتكوين قصيدة في البناء المماثل للجدول، يمكنك أن تعين كل STANZA و REFRAIN أن يكونا جدول، وكل VERSE أن يكون جدول صف يجب على ورقة النمط لكي تنشئ هذا التأثير أن تلك القواعد الثلاثة:

```
STANZA  { display: table }
REFRAIN { display: table }
VERSE   { display: table-row }
```

خصائص الارتفاع والعرض

يتم حساب الارتفاع الافتراضي للمربع يظهر فيه كل عنصر من خلال خليط من ارتفاعات محتويات العناصر. يتم حساب العرض الافتراضي لمربع كل عنصر من خلال خليط من عرض محتويات العناصر أو عرض المساحة المرئية على الصفحة أو الشاشة. يكون دائماً للعناصر المحولة وعناصر الجدول التي تحتوي على نص أبعاد محسوبة آلياً. بينما يجوز لمصممي ورقة النمط أن يغيروا تلك الافتراضات لعناصر مستوى الكتلة ويحل محلها عناصر محولة عن طريق قيم محددة لسمة خصائص:

١ .min-widt
.max-width
.min-height -
.max-height
.Height -
٦- .Width

تقوم خصائص min-height و min-width بتحديد أصغر البعاد التي يمكن للكائن أن يعرض بها تعد الخصائص القصوى حجم أقصى للمربع بغض النظر عن الحجم الكلي لمحتوياتها. يعتبر استعراض ويب له الحرية لتعديل حجم المربع في إطار تلك الحدود بينما، إذا تم تعيين height و width إذن فهما يحددان بالفعل حجم المربع.

```
STANZA  { width:  100px;
          Height: 100px }
```

خاصية تجاوز السمة

إذا تم تحديد حجم المربع بدقة عن طريق استخدام خصائص width و height فمن الممكن بسهولة أن تأخذ محتوياتها مساحة أكبر من المساحة الفعلية للمربع. تتحكم خاصية overflow في كيفية التعامل مع المحتوى الزائد. من الممكن تعيين تلك الخاصية لواحدة من الأربع قيم.

التي يتم عرضها، وكما تحتوي حاشية المربع على حاوية يظهر خلالها تخطيط الصفحات بين فواصل الصفحات خلاف المربعات الأخرى، لا يوجد بمربعات الصفحة أي حدود أو صفحات، يوجد فقط إلى هوامش.

تحدد قاعدة page@ كل صفحة في المستند. يمكنك استخدام إحدى خصائص الفئة المُقترحة للصفحة left: ,first أو right: لتحدد الخصائص المختلفة لمختلف الفئات من الصفحة.

عدم إدراك قاعدة page@ بمحتوى الصفحة التي تتضمن خطوطها، فهي لا تستطيع فهم المقياس ems و ex s تعد جميع وحدات القياس الأخرى مقبولة، بما في ذلك النسب المئوية. تعد النسب المئوية المستخدمة في إعدادات الهوامش، نسب مئوية لإجمالي مربع الصفحة. من الممكن أن يكون للهوامش قيم سلبية، تضع المحتوى خارج المساحة التي تقوم بالتشغيل المعهود بواسطة التطبيق أو الطابعة في معظم الحالات، تختصر المعلومة بسهولة إلى المساحة المرئية أو المطبوعة.

@import

تعمل قاعدة import@ على تضمين ورقة نمط خارجية محددة في ورقة من أجزاء صغيرة وسهلة الفهم. تستخدم أوراق النمط المستوردة ملحق css. على سبيل المثال، تستورد القاعدة التالية ملف css للشعر.

```
@import url(poetry.css);
```

كما يمكن لقاعدة import@ تحديد نوع الوسائط التي تلي اسم ورقة النمط. فإذا لم يتم تحديد أن نوع للوسائط، فإن قاعدة import@ تكون غير شرطية، وسيتم استخدامها مع كل أنواع الوسائط, على سبيل المثال، تستطيع القاعدة التالية استيراد ملف css للوسائط المطبوعة. وسيتم تطبيق التعريفات في ورقة النمط هذه على الوسائط المطبوعة.

```
@import url(printmedia.css) print;
```

تقوم القاعدة التالية باستيراد ملف css المتواصل الذي سيتم استخدامه لكلاً من شاشات الكمبيوتر و/أو شاشة:

```
@import url(continuous.css) tv, screen;
```

تكون للأوراق النمطية التي تم استيرادها من بين أوراق نمطية أخرى في رتبة مخصصة في التتالي أقل من ورقة النمط المستوردة على سبيل المثال، لنفرض أن قام شكسبير css بعمل VERSE نمطي في خط نيويورك، بينما قام shakeprint.css بتنميط VERSE في خط Times. إذا قام شكسبير css باستيراد shakeprint.css ، فإن بيوت الشعر سيتم جعلها نمطية في نيويورك بينما، إذا قام shakeprint.css باستيراد شكسبير css، فإن بيوت الشعر سيتم تنميطها في Times.

@media

تستخدم أنواع كثيرة من الوسائط في نقل المعلومات للقارئ ويكون لكل نوع من الوسائط أنماطـها وتنسيقاتها المخصصة. ولا يمكنك أن تحصل على متناول للمحادثة يقوم بقراءة شكســبير بنغمــة أحادية فهل يمكنك الآن؟ ولا يكون للنمط المائل شكل جيد مقبول على طرفية أحادية المسافة.

تمكنك CSS2 من تحديد أنماط مختلفة للعنصر الواحد المعروض في الوسائط على الشاشة إذا تم استخدام خط ليس رقعة، بينما تعد قراءة النص المكتوب على ورقة عامة أكثر سهولة إذا كــان مكتوب بخط الرقعة. تستطيع أن ترفق القواعد النمطية الموجهة لمتوسط واحد فقط فـــي قــاعدة @media التي تعرف تلك الوسائط. من الممكن أن يوجد العديد مـــن قواعـد @media فـــي المستند مثلما يوجد أنواع الوسائط الذي يتم تحديدها. على سبيل المثال، تقوم تلك القواعد بتنســيق SONNET مختلفة تعتمد على إذا ما كانت مطبوعة على ورق أو معروضة على الشاشة.

```
@media print {
  SONNET { font-size: 10pt; font-family: Times, serif }
}
@media screen {
  SONNET { font-size: 12pt;
         font-family: New York, Times New Roman, serif }
}
@media screen, print {
  VERSE { line-height: 1.2 }
}
```

تقوم أول قاعدتين بتعريف أنماط محددة لأنواع الوسائط المرئية على الشاشة والمطبوعة على التوالي وبما أن شاشات الكمبيوتر الحديثة أقل دقة من الطابعات الحديثة، فمن المهم جعل الخـــط على الشاشة أكبر من على النسخة المطبوعة، ومن المهم اختيار خط مصمم لاســـتخدامه علـــى الشاشة.

توفر القاعدة الثالثة المزيد من الأنماط التي يتم تطبيقها على كل نوع من أنواع هذه الوســائط. ولكي تصمم إرشادات نمطية لأنواع الوسائط المتعددة فورياً، تستطيع بسهولة سردهم بعد مصمـــم قاعدة @media وتفصلهم بفاصلة.

يستطيع المستعرضين الذين يدعموا CSS2 السماح لكاتب المستند بتوفير قواعد تنص علـــى كيفية عرض مستند ما لنوع معين من الوسائط. على سبيل المثال، ستقوم بتطبيق القواعد المختلفة عند إظهار مستند على الشاشة بطريقة أفضل من إرساله إلى الطابعة. يعرف CSS2 عشرة أنواع للوسائط وهما:

خاصية المؤشر

يعد المؤشر السهم/اليد/شريط إدراج/الرموز الأخرى التي تشير إلى موقع المؤشر على الشاشة. يعد المؤشر هو الممثل المرئي للموقع المنطقي للماوس الخاص بك الذي يظهر على المساحة المرئية لشاشة الكمبيوتر الخاص بك. تقوم خاصية cursor بتحديد المؤشر الذي يجب أن يظهر ببرنامج المستخدم عندما يحرك القارئ للماوس فوق كائن محدد. يسمح CSS2 بقيم المؤشرات السادسة عشر التالية:

١- auto: يختار المستعرض مؤشر يستند على المضمون الحالي تعد هذه القيمة الافتراضية.

٢- crosshair: مؤشر شعري سهل.

٣- default: مؤشر افتراضي معتمد على النظام الأساسي، غالباً ما يكون سهم.

٤- :hand: يد

٥- move: أسهم متقاطعة تشير إلى شيء سيتحرك.

٦- e-resize: سهم يشير إلى الشرق "فوق يعني الشمال"

٧- ne-resize: سهم يشير إلى الشمال الشرقي.

٨- nw-resize: سهم يشير إلى الشمال الجنوبي

٩- n-resize: سهم يشير إلى الشمال

١٠- se-resize: سهم يشير إلى الجنوب الشرقي

١١- sw-resize: سهم يشير إلى الجنوب الغربي

١٢- s-resize: سهم يشير إلى الجنوب

١٣- w-resize: سهم يشير إلى الغرب

١٤- text: I-beam

١٥- wait: ساعة التوقيت، كرة الشاطئ الزائدة أو الناقصة، الساعة الزجاجية أو الرموز الأخرى التي تشير إلى مرور الوقت.

١٦- help: علامة استفهام.

تستخدم القاعدة التالية خاصية cursor لتقول أن المؤشر اليدوي يجب أن يستخدم عندما يكون المؤشر فوق عنصر VERSE.

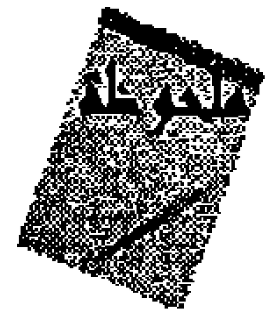
```
VERSE    { cursor: hand }
```

تستطيع أيضاً استخدام مؤشر مخصص يتم تحميله من ملف الصور عن طريق إعطـــاء URL للصورة عامةً ستقوم بتزويد المؤشرات بالتنسيقات المتعددة في قائمة منفصلة بالفاصلـــة، يكـــون أخرها، اسم نوع المؤشر. على سبيل المثال:

```
VERSE { cursor: url("poetry.cur"), url("poetry.gif"), text }
```

خاصية الألوان المرتبطة

يقوم CSS2 بتعريف اللون مثل قيم RGB في مسافة اللون الافتراضي القياسي للإنترنت "sRGB". تختلف الطريقة التي تقدم بها تلك الألوان من المستعرض لمستعرض أخر، ولكـــن توفـــر تلـــك المواصفات تعريف قياسي موضوعي غير غامض لمظهر اللون. يعد مستعرض الويـــب الـــذي يطابق القياسي الذي يشكل تصحيح للجاما على اللون المعرفة بواسطة مواصفات CSS2. تعـــوف SRGB الجاما التي تعرض ٢٠٢ تحت أحد الأحوال العرض. هـــذا يعنـــي أن لمعظـــم أجـــهزة الكمبيوتر، اللون الذي يتم إعطائه من خلال خصائص CSS2 يجب أن يتم تعديله لجامـــا مؤثـــرة معروضة بـــ ٢٠٢.



تكون الألوان المعرفة فقط بقواعد CSS2 مؤثرة من المتوقع للألوان التي تســتخدم في الصور أن تحمل معلومات تصحيح اللون الخلاصة بهم.

خاصية اللون

تحدد خاصية color الأمامي لمحتوى نص العنصر. يجوز أن تعطي كاسم لون حرفي مثـــل red أو قيمة RGB مثل CC0000#. تتضمـــن أســـماء الألـــوان "أو القيـــم" aqua وblack وblue وfuchsia وgray وgreen وlime وmaroon وnavy وolive وpurple وred وsilver وteal وwhite وyellow.

تطبق قواعد الأنماط التالية اللون الثلاث عناصر مستخدمة الطرق الثلاثة بأكملها لتعريف اللـــون. فهي تحدد قيمة عشرية FF0000# لعناصر AUTHOR جميع عناصر TITLE لتظـــهر اللـــون الأحمر، وجميع عناصر VERSE لتظهر rgb(255,0,0). تكون تلك القيم حمراء.

```
AUTHOR {  color: #FF0000}
TITLE   {  color: red}
VERSE   {  color: rgb(255,0,0) }
```

يمكن CSS2 كاتب المستند من تحديد أياً من تلك الأساليب إذا أمكن، استخدامها عندما لا يتم توافر أي خط مصمم مواصفات الخط، الذي تم إنشاء من بين سلسلة من الذين يقومون بوصف الخط، وتقوم بتعريف المعلومات المفصلة عن الخطوط التي تستخدم على الصفحة. ويتميز كل واصف للخط بجزء من المعلومات محددة عن الخط. ومن الممكن أن تتضمن تلك المواصفات URL للخط، الاسم العائلي للخط، وحجم الخط.

ينقسم من يقومون بوصف الخط إلى ثلاثة أنواع:

- الذين يقومون بتوفير رابطة بين استخدام ورقة نمط الخط ومواصفاته.
- الذين يقومون بتوفير URL لمكان الخط أو المعلومات وثيقة الصلة به.
- الذين يقومون بتوفير معلومات حرفية عن الخط.

تطبق قاعدة @font-face فقط على الخطوط المحددة داخل ورقة النمط. سوف تحتاج إلى إحدى مواصفات قاعدة @font-face لكل خط في ورقة النمط على سبيل المثال:

```
@font-face { font-family: "Comic Sans";
        src: url(http://metalab.unc.edu/xml/fonts/comicsans)}
@font-face { font-family: "Jester"; font-weight: bold;
           font-style: italic}
TITLE      { font-family: "Comic Sans"}
AUTHOR     { font-family: "Jester", serif}
```

بما أن البرنامج يقوم بقراءة ورقة النمط، فستتناول إيجاد مجموعة من القواعد التي تحدد كيف يتم معالجة كل عنصر تقوم ورقة النمط بتعيين جميع العناصر الخاصة TITLE لأصل الخط Comic Sans وفي الوقت نفسه تعيين جميع العناصر الخاصة AUTHOR لخط Jester وسيقوم تطبيق استعراض ويب الذي يدعم CSS1 بالبحث عن خط منصوص عليه لأصول خط Comic Sans وJester font وإذا لم تحدهم، فستستخدم الخط المنصوص عليه الافتراضي لأصل Comic، خط الرقعة المحدد المائل للخلف. سيتم تجاهل واصفي خط قاعدة @font-face. وسيكون برنامج CSS1 مستعداً لتخطي هذا الأمر دون احتساب أي خطأ.

وستقوم تطبيقات التي تدعم CSS2 باختبار قواعد واجهة @font-face في محاولة لتطابق مواصفات الخط بخطوط Comic Sans وjester. في المثال السابق، سيجد البرنامج المستعرض URL الذي يمكن عن طريقها تحميل خط Comic Sans إذا تم إيجاد خط Comic Sans في نظام العميل كان البرنامج سيستخدمه بدلاً من تحميل الخط وفي حالة Jester، يستخدم مستخدمي البرنامج إحدى قواعد المطابقة، أو قاعدة التركيب لإنشاء خط مشابه من الصفات المتوافرة. إذا كان استعراض ويب لا يستطيع إيجاد مطابقة قاعدة @font-face لعائلة الخط المحددة، كان سيحاول مطابقة الخطوط التي تستخدم القواعد الخاصة بـ CSS2.

كما يمكن CSS2 أي واصف للخط غير معروف، أو مفيد للمستعرض أن يتخطاه. مما يوفر وسائل للبناء الداخلي لزيادة الواصفين، في محاولة جاهدة لتحسين القواعد المستخدمة مثل استبدال الخط، مطابقته أو تركيبه.

@charset

يوجد هناك ثلاث طرق لتحديد مجموعة الأحرف التي يتم استخدامها في كابة ورقة النمط، وتأخذ تلك الطرق الأسبقية بالترتيب التالي:

١- معامل Charset HTTP في مجال "نوع-المحتوى"

٢- قاعدة @charset.

٣- السمات والخصائص المرتبطة بالمستند، مثل سمة HTML's Charset المستخدمة مع عنصر LINK.

من الممكن أن تحتوي كل ورقة نمط على قاعدة @charset الفردية ويجب أن تظهر قاعدة @charset في مستهل المستند، ولا يجوز مسبقاً بأي أخرف أخرى. أن بناء الجملة المستخدمة @charset هي:

```
@charset "character set name"
```

يجب أن يكون اسم مجموعة الأحرف المحددة في تلك العبارة اسم كما هو موصوف في مسجل IANA. تستطيع أن ترى قائمة جزئية لمجموعة أحرف في الجدول ٧-٧ في الفصل السابع ولكي تحدد أن ورقة النمط مكتوبة باللاتينية -١ فستكتب:

```
@charset "ISO-8859-1"
```

المرجع: يتم مناقشة مجموعات الأحرفُ بالتفصيل في الفصل السابع، اللغات الأجنبية والنص الغير روماني.

العناصر المُقترحة

يتم معاملة العناصر المُقترحة كعناصر في أوراق النمط ولكن ليس من الضرورة أن تكون عناصر محددة في مستند XML. عبارة عن مستخلصات من أجزاء معينة من المستند المعالج بعد تطبيق ورقة النمط، على سبيل المثال، أول خط من الفقرة لا يوجد العناصر المُقترحة حجم أحرف، ومن الممكن أن تظهر فقط مباشرة بعد موضوع محدد ورقة النمط. يقوم CSS2 بتحديد عنصرين مُقترحين جديدين after وbefore.

الجدول ١٣-٢
نظام الألوان الإضافية التي يجب استخدامها مع جميع خصائص اللون المرتبط

الكلمات الأساسية لنظام اللون	المواصفات
InactiveCaption	عنوان الويندوز والغير نشط
InactiveCaptionText	لون النص في عنوان غير نشط
InfoBackground	لون الخلفية لعناصر تحكم تعريف الأدوات
InfoText	لون النص لعناصر تحكم تعريف الأدوات
Menu	خلفية Menu
MenuText	النص في Menus
Scrollbar	مساحة أشرطة التمرير الرمادية
ThreeDDarkShadow	الظل الداكن لعناصر الأبعاد الثلاثة المعروفة
ThreeDFace	وجه اللون لعناصر الأبعاد الثلاثة المعروضة
ThreeDHighlight	اللون ذات إضاءة عالية لعناصر الأبعاد الثلاثة المعروضة
ThreeDLightShadow	اللون الخفيف لعناصر الأبعاد الثلاثة المعروفة "للأطراف التي تواجه مصدر الضوء"
ThreeDShadow	الظل الداكن لعناصر البعاد الثلاثة المعروضة
Window	خلفية ويندوز
WindowFrame	إطار ويندوز
WindowText	النص في الويندوز

خصائص الخط

غالباً ما تعد خصائص الخط في CSS1 كاملة. أما CSS2 فهي لا تضيف معظمهم تشمل التغييرات:

♦ إضافة خاصية font-size-adjust

- يعد عامل المقياس بين الأحجام المختلفة لخط الكلمات الأساسية xx-small و x-small و small و medium و large و x-large و(xx-large هو 1.2 وليس 1.5.
- تتمكن خاصية font-stretch من تقنين الأحرف.

خاصية تعديل حجم الخط

تعد مشروعية الخط عامة أقل اعتماداً على حجم الخط من اعتمادها على قيمة. ارتفاعها X. تعد قيمة أوجه الخط هي حجم الخط مقسوم على ارتفاع X كلما كان الرقم أعلى، كلما كان الخط مشروعية أكبر عندما يكون حجم الخط صغير. وكلما كانت القيمة الظاهرة أقل، كلما زادت قابلية الخط لأن يصبح غير قانوني لأنه يتقلص. عندما يكون المستعرضين بدائل "استبدالات" للخط المستقيم الأمامي المعتمد على وحدة حجم الخط، تتزايد بشدة قابلية عدم مشروعية الخط المتبقي. تتحكم خاصية font-size-adjust القيمة الظاهرة للعناصر التي تحتفظ x-height للخط المختار الأول في الخط المستبدل عند استخدام خاصية font-family.

توفر خطوط Verdana و Times New Roman مثال جيد لهذه القضية المشروعة تكون Verdana قيمة ظاهرة ٥٨.، بينما تكون القيمة الظاهرة Times New Roman ٤٦. ولهذا ستظل Verdana مشروعة بحجم أصغر من Times New Roman ولكن من الممكن أن تظهر كبيرة جداً إذا تم استبدالها مباشرة Times New Roman التي لها نفس حجم الخط.

إذا كانت قيمة خاصية font-size-adjust يكون none، لا يتم الحفاظ على ارتفاع X للخط فإذا تم تحديد أي رقم، تقوم القيمة بتعريف القيمة الظاهرة الخط الأول المختار، وتقوم بتوجيه البرنامج لقياس الخط المستبدل تباعاً قد يساعدك تلك النظام على فرض المشروعية غير جميع الأنظمة الأساسية وجميع التطبيقات المدعمة تستخدم القاعدة التالية خاصية font-size-adjust للمحافظة على مشروعية الخطوط عند تنفيذ نظام من الأحجام.

```
VERSE  { font-size-adjust: "85.";  }
        font-family: ;Verdana, Times New Roman,
                    Helvetica, Arial " ; }
AUTHOR { font-size-adjust:"46." }
        font-family: "Times New Roman, Goudy Old Style,
                    serif, fantasy"; }
```

خاصية الخط المتعدد

تتحكم خاصية font-stretch في تقنين أحرف الخط، وهذا يعني كمية المسافات الموجودة بين الحرفين في الخط يوجد أثنى عشر للكلمات الأساسية المشروعة لهذه الخاصية:

:lang()

تحدد الفئة المُقترحة ()lang العناصر التي لها لغة محددة. ويتم ذلك عاماً في XML بواسطة سمة xml:lang و/أو سمة وضع التغييرات البرمجية لإعلان تعريف XML تقوم القاعدة التالية بتغيير اتجاه جميع عناصر VERSE المكتوبة باللغة اليهودية ليتم قراءتها من اليمين إلى الشمال، بدلاً من الشمال إلى اليمين:

```
VERSE:lang(he) {direction: "rtl" }
```

:right, :left, :first

تعد الفئات المُقترحة right وleft: وfirst: مطبقة فقط على قاعدة @page rule فهي تمكنك من تحديد الأنماط المختلفة للصفحة الأولى من المستند، وللصفحات اليسرى للمستند "عامة مرقمة زوجي" وللصفحات اليمنى للمستند "عامة مرقمة فردي". على سبيل المثال، تقوم تلك تقوم القواعد بتحديد الهوامش الكبيرة جداً.

```
@page:right  { margin-top: 5cm;
             margin-bottom: 5cm;
             margin-left: 7cm;
             margin-right: 5cm }
@page:left   { margin-top: 5cm;
             margin-bottom: 5cm;
             margin-left: 5cm;
             margin-right: 7cm }
@page:first  { margin-top: 10cm;
             margin-bottom: 10cm;
             margin-left: 10cm;
             margin-right: 10cm }
```

الوحيدة التي يمكنك تعيينها في القاعدة لتلك الفئات المُقترحة هي الخصائص الخاصة بالهامش.

تنسيق صفحة

يقوم محدد @page بالإشارة إلى صفحة وهو معتاد على تعيين الخصائص المطبقة على الصفحة نفسها بدلاً من عنصر XML الفردي على الصفحة. يوجد لكل صفحة في المستند مجموعة متنوعة من الخصائص التي تطبق عليها، بما في ذلك حجم الصفحة، الاتجاه، الهوامش، وفواصل الصفحات. تتلو هذه الخصائص أي عنصر موجود على الصفحة. ويمكن للفئات المُقترحة الاختيارية أن تحدد خصائص مختلفة للصفحة الأولى، الصفحات ذات الأوجه اليمنى والصفحات ذات الأوجه اليسرى

يقوم CSS2 بالافتراض المعقول أن تكون الصفحات مستطيلة. بتوافر هذا الافتراض يجـــوز للصفحة أن تضع خصائص المربع المألوفة لك في نظام CSS1 بما في ذلك الهوامش والحجـــم. بينما، لا يوجد لمربع الصفحة أي حدود أو مسافة فارغة بينها وبين البيانات، لأنهم من الطبيعـــي أن يكونوا موجودين خارج الصفحة الحقيقية.

خاصية الحجم

في قاعدة page@ تحدد خاصية size ارتفاع وعرض الصفحة. يمكنك تعييـــن size بواحـــد أو اثنين من الأطوال المطلقة أو كواحـــد مـــن الكلمـــات الأساسـية الأربعـــة ,auto, portrait landscape وinherit فإذا تم إعطاء طوال واحد فقط فإن الصفحة ســـتكون مربعـــة. فـــإذا تم إعطاء الأطوال كلها، فإن الأولى هي عرض الصفحة، والثانية هي طولها على سبيل المثال،

```
@page { size: 8.5in 11in }
```

تأخذ إعدادات auto حجمها آلياً على الورقة أو الشاشة الهدف. أما landscape فهي تفرض على المستند أن يكون منسق لتناسب الصفحة الهدف، ولكن بجوانب أفقيـــاً طويلـــة كمـــا تنسق إعدادات portrait المستند ليناسب حجم الصفحة ذات الهدف الافتراضي، ولكن بجوانب رأســـية طويلة.

خاصية الهامش

تتحكم خاصية margin في هوامش الصفحة ---- المساحات المستطيلة الموجودة في الجوانـــب الأربعة التي لا يكون مطبوع عليها أي شيء تستخدم هذه الخاصية كاختزال لتعييـــن خصـــائص margin-top وmargin-bottom وmargin-right وmargin-left كلاً على حدة تعـــد هـــذه الخصائص نفس الخصائص الخاصة بالمربعات في CSS1 على سبيل المثال، تصف هذه القـــاعدة ٨,٥ في ١١ بوصة صفحة مع بوصة واحد من الهوامش في جميع الجوانب.

```
@page { size: 8.5in 11in; margin: 1.0in }
```

خاصية الترميز

يوفر لنا CSS2 لتضع العلامات على صفحة ترسم أين تتم قطع الأوراق و/أو كيف يتم محـــاذاة الصفحات. تظهر تلك العلامات خارج مربع للصفحة يصبح مربع الصفحة بسهولة أكثر مســـاحة مرئية في المستند الذي قد يتأثر بقاعدة page@ إذا كنت تنظر إلى جزء من الورقة مطبـــوع 8 1/2 "x11" سيكون مربع الصفحة كل شيء داخل المنطقة المطبوعة على تلك الورقة، وهو مـــا نعتقد طبيعياً أنه المسافة داخل الهوامش المطبوعة. يتحكم البرنامج في أداء العلامات، التي يتـــم

١- baseline: وهم محاذاة الأساس للمربع المحول بالخط الأساسي

٢- sub: محاذاة الخط الأساسي للمربع المحول لمكان المنخفضات داخل مربع الكتلة الأصلي.

٣- Super: ترفع الخط الأساسي للمربع المحول إلى مكان المنخفضات داخل المربع الأصلي.

٤- top: محاذاة قمة المربع المحول بقمة الخط.

٥- middle: محاذاة النقطة الوسطى للمربع المحول بالخط الأساسي middle لمربع الكتلة، بالإضافة إلى (+) نصف ارتفاع -x.

٦- bottom: محاذاة أسفل المربع المحول بقاع الخط.

٧- text-top: محاذاة قمة المربع المحول بقمة أصل خط العنصر.

٨- text-bottom: محاذاة أسفل المربع المحول بأسفل أصل خط العنصر.

تستطيع أيضاً تعيين خاصية vertical-align للنسبة المئوية التي ترفع "القيمة الإيجابية" أو خفض "القيمة السلبية" المربع بواسطة النسب المئوية لارتفاع الخط. تعد قيمة ٥% تماماً مثل الكلمات الأساسية baseline. أخيراً، تستطيع تعيين vertical-align للطول المستحيل الذي سيرفع أو يخفض المربع بواسطة المساحة المحددة. تعد قيمة ٥ سم تماماً مثل الكلمة الأساسية baseline.

المربعات

عندما تستخدم CSS لتنسيق المستند ومحتوياته تحتاج إلى التفكير في نهاية المربعات بالحدود والأبعاد التي تحمل محتويات العنصر. تعد تلك المربعات مرصوفة مع بعضها وتلف حول بعضها البعض حتى يتم محاذاة محتويات كل عنصر في شكل مرتب، يستند على قواعد أوراق النمط. يضيف CSS2 خصائص مخطط تفصيلي جديد في أماكن مطلقة على الصفحة، في مربع أخر، أو في الويندوز.

خصائص المخطط التفصيلي

تجعل CSS2 من الممكن إضافة مخططات تفصيلية للكائنات يشبه المخطط التفصيلي فوق المربع. ويضاف عرضها لعرض المربع بالإضافة إلى ذلك، إذا كان عنصر CSS2 غير مستطيل "غير قابل"، ستكون المخططات التفصيلية التي حولها غير مستطيلة أيضاً وبما أن المخططات التفصيلية ليست بالضرورة مستطيلة، لا تستطيع أن تعين اليسار، يمين، قمة وأسفل المخطط التفصيلي على حدة يمكنك فقط أن تؤثر على المخطط التفصيلي بأكملها على الفور.

خاصية نمط المخططات التفصيلية

تقوم خاصية outline-style بتعيين نمط المخطط التفصيلي للمربع بأكمل. تقوم تلك الخاصية بنفس وظيفة خاصية border-style property في CSS1، وله نفس القيم الإحدى عشرة الممكنة التي لها نفس المعاني:

١- none: بلا خط
٢- hidden: خط غير مرئي لا يزال يأخذ مسافة
٣- dotted: خط منقط
٤- dashed: خط متقطع
٥- solid: خط صلب
٦- double: خط صلب متضاعف
٧- grooved: خط يظهر كأنه مرسوم على الصفحة
٨- ridge: خط يظهر وكأنه قادم على الصفحة
٩- inset: الكائن بأكمله "وليس فقط المخطط التفصيلي"
١٠-outset: الكائن بأكمله "وليس فقط خط المخطط التفصيلي"
١١-inherit: يظهر كأنه مدفوع إلى خارج المستند

نعين تلك القواعد الثلاثة أنماط المخطط التفصيلي لعناصر TITLE وAUTHOR وREFRAIN:

```
TITLE   { outline-style: solid }
AUTHOR  { outline-style: outset }
REFRAIN { outline-style: dashed }
```

خاصية عرض المخطط التفصيلي

تكمل خاصية outline-width مثل خصائص margin-width وborder-width التي تم مناقشتها في الفصل ١٢ فهي تعين عرض المخطط التفصيلي للمربع الذي يستخدم إما طول غير مسجل أو واحدة من تلك الكلمات الأساسية الثلاثة:

١- thin: تقريباً ٠,٥ إلى ٠,٧٥ نقاط.
٢- medium: تقريباً نقطة واحدة.
٣- thick: تقريباً من ١,٥ إلى نقطتين.

على سبيل المثال، تقوم هذه القاعدة بعمل مخططات تفصيلية STANZA بمخطـــط تفصيلـــي سميك VERSE بمخطط تفصيلي رفيع.

```
STANZA    { outline: thick }
VERSE     { outline: thin }
```

خاصية لون المخطط التفصيلي

تعين خاصية لون outline-color لون المخطط التفصيلي لمربع العنصر. عامة، ويتعين هذا إمـا لاسم لون مثل الأحمر أو لون RGB مثل #FF0000. بينما يجوز أن يكون لها أيضاً قيمة الكلمـة الأساسية invert التي تعكس لون البكسل على الشاشة. "الأسود يصبح أبيض والعكس صحيـــح" على سبيل المثال:

```
TITLE  {  outline-color: #FFCCCC;
          outline-style: inset;
          outline-width: thick}
AUTHOR {  outline-color: #FF33CC}
VERSE  {  outline-color: invert}
```

خاصية اختزال المخطط التفصيلي

تعد خاصية outline خاصية اختزال تعيين عرض، لون نمط المخطط التفصيلي لكل الأطـــراف الربعة لمربع ومحتوى. على سبيل المثال:

```
STANZA    { outline: thin dashed red }
VERSE     { outline: inset }
```

خصائص الموضع

يوفر لنا CSS2 درجة مذهلة من التحكم في موضع كل كائن في المســـتند. تســـتطيع أن تضـــع كائنات محددة أو أنواع محددة للكائنات في طبقات. قد تتحرك كل طبقة مســتقلة عــن الطبقــات الأخرى. تحدد خاصية position كيفية ترتيب الكائنات وقد يكون لها واحدة من قيـــم الكلمــات الأساسية الأربعة وهم:

١- static: تخطيط افتراضي.

٢- relative: الكائنات التي تم إزاحتها من موضعهم الثابت.

٣- absolute: توضع الكائنات بمكان محدد مرتبط بالمربع الذي قد احتواهم.

٤- fixed: توضع الكائنات عند نقطة محددة في الويندوز أو على الصفحة.

الموضع النسبي

بتخطيط المستند، يختار المنسق مواضع البنود طبقاً للانسياب الطبيعي للكائنات والنص. تعد هذه بالضرورة التنسيق الثابت الافتراضي للكائنات المستخدمة بواسطة معظم مصممي المستند ويعد اكتمال هذا، يجوز إزاحة الكائنات نسبياً إلى موضعهم الحالي. يعد هذا التعديل في موضع الكائن معروف بالموضع النسبي. باستخدام الموضع النسبي لا يوجد لتعديل موضع الكائن أي تأثير على الكائنات التي تليها. لذلك فمن الممكن تراكب المربعات. إذ أن المربعات الموضوعة نسبياً تحتفظ بجميع الأحجام والمسافات الانسيابية الطبيعية.

تستطيع أن تنتج كائن موضوع نسبياً عن طريق إعداد خاصية position إلى relative. وسيتم التحكم في إزاحتها عن طريق خصائص left وright وtop وbottom تستطيع عند تغيير تلك الخصائص JavaScript أن تحرك الكائنات والطبقات على مستندات. تستطيع أن تجعل الصور والنصوص تتحرك وتظهر أو تختفي أو تتغير أثناء تنفيذ الإجراءات والنصوص تتحرك وتظهر أو تختفي تلك القاعدة عنصر TITLE ٥٠ بكسل للأعلى و٦٥ بكسل لليسار من المكان الذي من الطبيعي أن تكون به.

```
TITLE   { position: relative;  top: 50px;  left: 65px}
```

الموضع المطلق

يتم وضع عنصر الموضع المطلق بالإشارة إلى الكتلة التي تحتوي عليها فهي تنشئ كتلة ذات محتوى جديد للمربعات التي تحتويها. لانسياب محتويات العناصر الموضوعة المطلقة حول المربعات الأخرى. قد يسبب يقودهم هذا إلى عدم توضيح محتويات المربعات الأخرى المعروضة في المستند لا يوجد للعناصر الموضوعة المطلقة أي تأثير على الانسيابية المرتبطة المتتالية، إذاً فالعناصر التي تتبع العنصر الموضوع المطلق تتصرف وكأنها ليست موجودة هناك. على سبيل المثال، تضع هذه القاعدة الركن اليساري الأعلى لعنصر AUTHOR ٦٠ بكسل للأسفل، و١٤ بكسل على يمين الركن اليساري الأعلى للمربع الذي كانت توجد بداخله "يحتويها".

```
AUTHOR  { position: absolute;  top: 60px;  left: 140px }
```

الموضع الثابت

توضع العناصر التي لها موضع ثابت منسقة نسبياً للويندوز أو الصفحة التي يتم عرضها عليها. إذا كنت تشاهد مستند متكون من وسائط مستمر فإن المربع الثابت على الوسائط المرقمة، فستظهر دائماً في نهاية كل صفحة. يمكنك هذا من وضع تذييل أو رأس للصفحة على المستند، أو

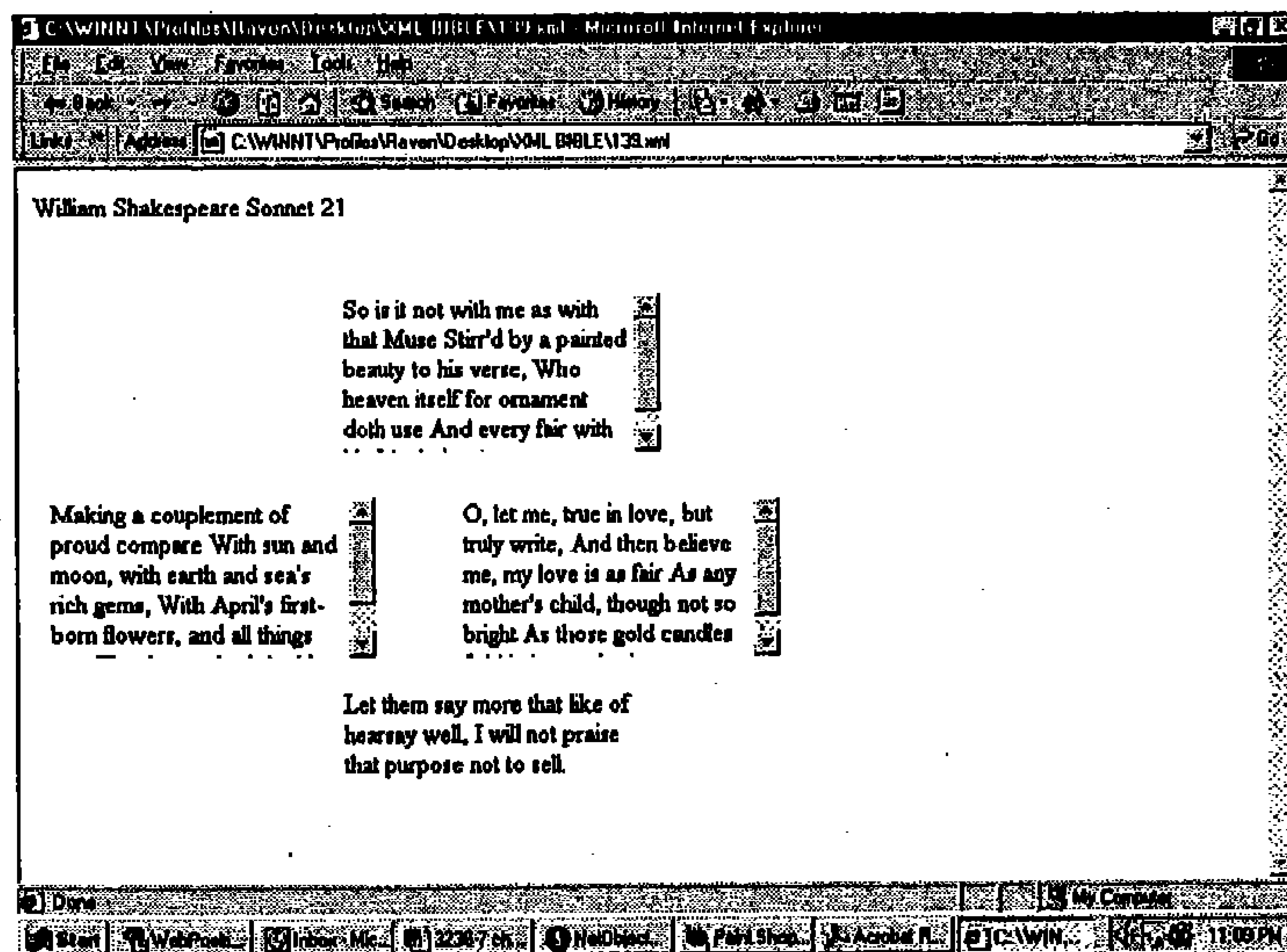
توقيع في نهاية سلسلة من حروف الصفحة الواحدة على سبيل المثال، تضع تلك القاعدة الركن اليساري الأعلى لعنصر REFRAIN ٣٠٠ بكسل للأسفل و ١٤٠ بكسل على يميل الركن اليساري الأعلى للويندوز والمعروضة فيها أو للصفحة المطبوعة عليها.

```
REFRAIN  { position: fixed;  top: 300px;  left: 140px}
```

خاصية العناصر المرصوفة Z-Index

تتحكم خاصية Z-Index في الترتيب المرصوفة للمربعات الموضوعة ولتغيير قيمة Z-Index الافتراضية، تقوم بتعيين Z-Index لعدد صحيح مثل رقم اثنين. أما الكائنات التي لها قيم Z-Index كبيرة فتوضع على قمة الكائنات بقيم Z-Index صغيرة تعد مسألة ظهور الكائنات على الأزرار وظيفته خصائص الخلفية للكائن الذي يوجد على قمتهم. إذا كانت الخلفية شفافة، من المحتمل ظهور على الأقل بعضاً مما يوجد تحتها.

تعد التعليمات البرمجية ١٣-٢ ورقة نمط تستخدم الموضع المطلق شكسبير. تظهر النتيجة في الشكل ١٣-٣ بالطبع هي ليست لطيفة مثل الإصدار الذي يسمح للمستعرض بتخطيط القصيدة من الممكن استخدام الموضع المطلق بحذر كبير. في الحقيقة، لقد أوصيت بها فقط للوسائط المطبوعة حيث يمكنك توزيع الورقة التي تخرج من طابعتك بدلاً من الملفات الإلكترونية.



الشكل ١٣-٣ باستخدام الموضع المطلق المرتب بواسطة z-index، تستطيع التحكم في الترتيب المرصوف لمربعات النص.

تعليمات البرمجة ١٣-٢: قصيدة شكسبير مع ورقة نمط z-index

```
#st1    { position: absolute;
         top: 160px;
         left:200px;
         height: 100px;
         width:200px;
         overflow: auto;
         z-index: 2}
#st2    { position: absolute;
         top: 210px;
         left:50px;
         height: 100px;
         width:200px;
         overflow: auto;
         z-index: 3}
#st3    { position: absolute;
         top: 210px;
         left:250px;
         height: 100px;
         width:200px;
         overflow: auto;
         z-index: 4}
REFRAIN { position: absolute;
         top: 300px;
         left:200px;
         height: 100px;
         width:200px;
         overflow: auto;
         z-index: 5}
```

الترقيم الآلي وأجهزة العد

يمكنك CSS2 من أن تنتج آلياً بعض المحتويات على سبيل المثال، تستطيع استخدام ورقة النمط لتنشئ مخططات تفصيلية التي من المحتمل أن تكون مسافة بادئة مع أنظمة الترقيم المختلفة لكل مستوى من المخطط التفصيلي.

تضيف خاصية counter-increment واحدة لجهاز العد. كما تقوم خاصية content بإدراج القيمة الحالية لجهاز العد المسمى باستخدام دوال القيم إما counter(id) أو counter(id, list-style-type) أخيراً، تعين خاصية counter-reset عد تنازلي إلى الصفر.

على سبيل المثال، دعنا نفترض أنك تريد أن ترقم كل VERSE في الشعر يبدأ من واحد ولكن تعيد تعيين الحساب في كل STANZA جديد وREFRAIN. تستطيع أن تفعل ذلك بالقواعد التالية:

```
VERSE        {counter-increment: verse-num}
STANZA       {counter-reset: verse-num}
REFRAIN      {counter-reset: verse-num}
VERSE:before {content: counter(verse-num) }
```

تستطيع أن تعيد التعيين تنازلياً لرقم غير الصفر عن طريق تجديد العدد الصحيح الذي يعاد تعيينه بعد اسم جهاز العد في counter-reset. على سبيل المثال، إعادة تعيين جهاز العد إلى عشرة.

```
VERSE        {counter-reset: verse-num -10}
```

تستطيع أيضاً أن تزودها بواسطة عدد صحيح مختلف عن واحد عن طريق تحديده في counter-increment بعد اسم جهاز العد. على سبيل المثال:

```
VERSE        {counter-increment: verse-num -1}
```

أخيراً، يمكن لخاصية content أن يكون لها أكثر من جهاز عن واحد، ومضمون إضافي بالإضافة إلى أجهزة العد. على سبيل المثال، تقوم تلك القواعد بترقيم بيوت الشعر بالشكل ١٠١، ١٠٢، ١٠٣ .... ٢٠١، ٢٠٢، ٢٠٣ حيث يشير الرقم الأول إلى المقطع الشعري ويشير الثاني إلى البيت الشعري.

```
VERSE        {counter-increment: verse-num}
STANZA       {counter-reset: verse-num}
STANZA       {counter-increment: stanza-num}
REFRAIN      {counter-reset: verse-num}
REFRAIN      {counter-reset: stanza-num 0}
VERSE:before {content:
               counter(stanza-num) "." counter(verse-num) }
```

ونحن غير محدودين بالأرقام الأوروبية أيضاً تستطيع أن تطرح جدال ثاني لوظيفة counter() لتحديد شكل رقمي مختلف. تتضمن الأشكال المتوفرة على disc, circle, square, decimal, decimal-leading-zero, lower-roman, upper-roman, lower-greek, lower-alpha, lower-latin, upper-alpha, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, and katakana-iroha. على سبيل المثال، لكي تقوم بترقيم بيوت الشعر التي تستخدم أرقام يابانية في hiragana، يمكنك أن تكتب.

```
VERSE:before {content: counter(stanza-num, hiragana)
               "." counter(verse-num, hiragana) }
```

أوراق النمط السمعية

يستخدم المستخدمين الضعاف الواضحين بالفعل برنامج خاص لقراءة صفحات ويب ففي المستقبل، يكون لمثال هذا الاستخدام قابلية التوسع للأشخاص ذو وجهة نظر معينة المستعرضين للويب بينما يتكلمون على الخلايا. التليفزيونية، قيادة سيارتهم، غسيل الصحون، والقيام بالأنشطة الأخرى حيث يتم توجيه الأيادي والأعين لناحية أخرى يدعم CSS2 الخصائص الجديدة ليصف كيفية قراءة العناصر بصوت عالي بالإضافة إلى كيفية طبعها أو ظهورها على الشاشة. ويتم مناقشة الخصائص الجديدة في الأقسام التالية من الكتاب. تعليمات البرمجة ١٣-٣ من ورقة نمط سمعية تقوم بتعريف طرق محددة للتحدث بالمعلومات الموجودة في عناصر play-related XML المشتركة.

تعليمات البرمجة ١٣-٣: ورقة نمط سمعية لمسرحية أو قصيدة

```
, ACT, SCENE {
     voice-family: narrator;
     stress: 20;
     richness: 90;
     cue-before: url("ping.au")
  }

.narrator { pause: 20ms;
         cue-before: url("pop.au");
         cue-after: url("pop.au");
```

```
        azimuth: 30deg;
        elevation: above }

ACT   { pause: 30ms 40ms } /* pause-before: 30ms;
                      pause-after: 40ms */

SCENE { pause-after: 10ms } /* pause-after: 10ms */

SCENE { cue-before: url("bell.aiff");
      cue-after: url("dong.wav") }

MOOD.sad       { play-during: url("violins.aiff") }
MOOD.funereal { play-during: url("harp.wav") mix }
MOOD.quiet     { play-during: none }

LINE.narrator      { azimuth: behind } /* 180deg */
LINE.part.romeo     { voice-family: romeo, male }
LINE.part.juliet   { voice-family: juliet, female }
LINE.part.hercules { azimuth: center-left }
LINE.part.richard  { azimuth: right }
LINE.part.carmen    { volume: x-soft }
LINE.part.muse1     { elevation: 60deg }
LINE.part.muse2     { elevation: 30deg }
LINE.part.muse3     { elevation: level }
```

خاصية الحديث

تحدد خاصية speak إذا ما كان النص سيتم تقديمه سمعياً أم لا، وإذا كان كذلك فكيف. إذا كان speak قيمة normal، يتم التكلم بالنص باستخدام، أفضل تراكيب الحديث المتوفرة. إذا كان speak قيمة spelled out، يتم نطق النص حرف بحرف، الذي قد يعد مفيداً للكلمات غير المعتادة أو الغريبة التي من المحتمل ألا يستطيع محللي الحديث تناولها. القيمة الافتراضية none "على سبيل المثال، قدم فقط المحتوى مرئياً وأنسى أمر تحليل الحديث".

خاصية حجم الصوت

تتحكم خاصية volume في متوسط حجم الصوت لصوت المتكلم لمحللي الحديث. تعد هذه قيمة متوسطة لموجه الصوت المخاطرة للصوت، ولكنه فقط متوسط يصل صوت متغير عالي بحجم صوت ٥٠ إلى الذروة عند ٧٥. يعد الحد الأدنى لحجم الصوت صفر. أما الحد الأقصى لحد الصوت فهو مائة. من الممكن أيضاً استخدام قيم النسب المئوية مثل استخدام أياً من تلك الكلمات الأساسية الستة.

١- silent: بلا صوت.

٢- x-soft: صفر الحد الأدنى لحجم الصوت المسموع.

٣- soft: تقريباً ٢٥.

٤- medium: تقريباً ٥٠.

٥- loud: تقريباً ٧٥.

٦- x-loud: الحد الأقصى لمستوى السمع المربع هو ١٠٠.

خصائص الإيقاف المؤقت

يعد الإيقاف المؤقت مساوي سمعي للفاصلة. من الممكن استخدامهم لعرض مأساة، أو فقط للمساعدة على فصل صوت متكلم واحد عن الآخر. يتم تعيينهم في CSS2 عن طريق خصائص pause, pause-before و pause-after.

تحدد خاصية pause-before طول الوقت التي يجب أن يقوم مؤلفي الكلام بالإيقاف المؤقت قبل التكلم عن محتويات العنصر تحدد خاصية pause before طول الوقت التي يجب على مؤلفي اللغة أن يقوموا بالإيقاف المؤقت بعد التكلم عن محتويات العنصر. يمكن تعيين ذلك كوقت مطلق أو كنسبة مئوية لخاصية نسبية الكلام تعد خاصية pause اختزال لإعداد كلاً من pause-before و pause-after. عندما يتم تجهيز القيمتين، يتم تطبيق الأولى على pause وتطبق الثانية على pause-after. عندما يتم إعطاء قيمة واحدة فقط، فيتم تطبيقها على كلا الخصائص. على سبيل المثال:

```
SCENE { pause-after: 10ms }

/* pause-before: 20ms; pause-after: 20ms */
.narrator { pause: 20ms }

/* pause-before: 30ms; pause-after: 40ms */
ACT   { pause: 30ms 40ms }
```

خصائص المساعدة

تعد المساعدات مساعدات سمعية تأخذ المستمع إلى حدث يحدده على وشك أن يحدث أو قد حدث لتوه تحدد كل خاصية مساعدة الـ URL لملف صوت سيتم قراءته قبل أو بعد تحدث العنصر. تقوم خاصية cue-after بقراءة الصوت قبل قراءة العنصر. إما خاصية cue-after فهي تقرأ الصوت بعد قراءة العنصر.

تعد خاصية cue اختزال لإعداد كلاً من cue-before وcue-after وعندما يتم تجهيز القيمتين، تطبق الأولى cue-after وعندما يتم إعطاء قيمة واحدة فقط، فهي تطبق على كلاً الخصائص على سبيل المثال:

```
ACT, SCENE { cue-before: url("ping.au") }
.narrator  { cue: url("pop.au") }
SCENE      { cue-before: url("bell.aiff");
           cue-after: url("dong.wav") }
```

خاصية Play-During

تحدد خاصية play-during الصوت الذي سيتم قراءته في الخلفية أثناء حديث محتويات العنصر. تعد قيمة الخاصية URL لملف الصوت. يمكنك أيضاً بإضافة كلمة أ, أكثر من الكلمات الأساسية mix وrepeat إلى القيمة. Mix تخير مؤلفي المحادثة أن يقوموا بخلط أصل صوت play-during. أما قيمة repeat فهي تخبر مؤلفي اللغة أن يجعلوا الصوت في حلقات مستمرة إلى أن يتكلم العنصر بأكمله. تكون القيمة الافتراضية none.

خصائص Spatial

تحدد خصائص spatial المكان الذي يأتي منه الصوت على سبيل المثال، يمكن أن يكون عندك مستند يقرأ عليك من على بعد ٣٠ قدم أو ١٠٠ قدم بالطبع يتم تحديد هذا بواسطة قدرات مؤلفي المحادثة والأجهزة السمعية. وبما أنك لا تستطيع أن تحدد سابقاً رقم ومكان المتحدثين الفعليين بواسطة قارئ المستند فإن تلك الخصائص تعرف بسهولة النتيجة النهائية المرغوب فيها. وككاتب للمستند، فلا تستطيع حقاً أن تفرض على الصوت أن يأتي من أي اتجاه محدد، ولا تستطيع بالضبط أن تضمن أن القارئ لديه شاشة ملونة.

خاصية Azimuth

تتحكم خاصية Azimuth في الزاوية الأفقية الذي يبدو أن الصوت يأتي منـــها، عندمـــا تســـتمع لجهاز لصوت شكل مكبرات صوت استريو جيدة، يبدو أنك تسمع مرحلة صوت جـــافي، يمكـــن استخدام خاصية Azimuth مع هذا النوع من نظام الاستريو لتنشئ زوايا الصوت الذي تســـمعه. عندما تضيف نظام صوتي. محيط مكتمل باستخدام إما سماعات بأذنين أو شاشة إعداد لمســـــرح بخمس مكبرات الصوت عندئذ تصبح خاصية azimuth واضحة جداً.

تحدد azimuth بزاوية بين ٣٦٠ و٣٦٠. تعني قيمة 0deg أن الصوت يوجد مباشرة أمـــام المستمع "مثل 360deg و360deg" وتعني قيمة 180deg أن الصوت بعد مباشرة خلف المستمع "في CSS فإن مصطلحات deg تحل محل رمز الدرجة ( ) الأكثر شيوعاً". تحسب الزوايا فـــي اتجاه الساعة على يمين المستمع. يمكنك أيضاً استخدام تلك الكلمات الأساسية التسعة لتحدد زاوية azimuthal.

١- center: درجة صفر.

٢- center-right: ٢٠ رجة.

٣- right: ٤٠ درجة.

٤- far-right: ٦٠ درجة.

٥- far-right: ٩٠ درجة.

٦- right-side: ٢٧٠ درجة.

٧- far-left: ٣٠٠ درجة.

٨- left: ٣٢٠ درجة.

٩- center-left: ٣٤٠ درجة.

يمكنك إضافة الكلمة الأساسية behind أياً من تلك القيم لتعيين الموضع إلى 180deg ناقص القيمة العادية. على سبيل المثال، تعد left behind تماماً مثل درجة ٢٢٠ و١٤٠= درجة ٣٢٠- درجة ١٨٠.

تحرك قيمة leftwards الصوت ٢٠ درجة إضافية إلى اليسار ولها علاقة نســـبية بالزاويـــة الحالية. يتم فهم هذا بسهولة مثل تحويل عداد الصوت عكس اتجاه عقارب الساعة. لذلك حتـــى إذا كان الصوت بالفعل خلف المستمع، فسيستمر الحركة "لليسار" حول الذاكرة. كما تحـــرك قيمـــة rightwards الصوت ٢ درجة إضافية إلى اليمين (في اتجاه عقارب الساعة) في اتجاه الزاويـــة الحالية.

خاصية الرفع Elevation

تتحكم خاصية elevation في الارتفاع الظاهر لمكبر الصوت فوق موضع المستمع يحدد الرفع كزاوية بين ٩٠ و ٩٠-. ومن الممكن أيضاً أن تعطى كواحدة من الكلمات الأساسية الخمسة:

١- below: ٩٠ درجة.
٢- level: ٠ درجة.
٣- above: ٩٠ درجة.
٤- higher: ١٠ درجات فوق الرفع الحالي "مفيد مع الوراثة".
٥- lower: ١٠ درجات قمت الرفع الحالي "مفيد مع الوراثة".

خصائص مميزات الصوت

من الممكن التحكم في المميزات لصوت المؤلف بواسطة تعديل معدل المحادثة، أصل الصوت المستخدم، الخطوة، وتنسيق الصوت.

خاصية معدل المحادثة

تحدد خاصية speech-rate معدل حديث مؤلفي المحادثة برقم مقرب له متوسط حجم الكلمة كل دقيقة. تستطيع تجهيز عدد صحيح أو واحد من الكلمات الأساسية الخمسة.

١- x-slow: ٨٠ كلمة في الدقيقة.
٢- slow: ١٢٠ كلمة في الدقيقة.
٣- medium: من ١٨٠ إلى ٢٠٠ كلمة في الدقيقة.
٤- fast: ٣٠٠ كلمة في الدقيقة.
٥- x-fast: ٥٠٠ كلمة في الدقيقة.

تستطيع أيضاً أن تستخدم الكلمة الأساسية faster لتضيف ٤٠٠ كلمة في الدقيقة لمعدل العنصر الأصلي أو كلمة slower لتنقص ٤٠ كلمة في الدقيقة من معدل أصل العنصر.

خاصية أصل الصوت

تعد خاصية voice-family مفصولة بفاصلة، قائمة ذات أولوية لأسماء أصل الصوت التي تختار الصوت المستخدم لقراءة نص المستند. إنها تشبه خاصية font-family التي تمت مناقشتها في الفصل ١٢، ولكن باستخدام الأصوات بدلاً من المحرفات.

تتضمن قيم الأصوات الناتجة male وfemale وchild. تعد الأسماء المحددة مختلفة ومتعددة مثل أسماء الخطوط وتتضمن Agnes وBruce, Good News, Hysterical, Victoria, Whisper، وأسماء أخرى كثيرة. يجب اقتباس تلك الأسماء إذا لم تطابق قواعد بناء السمة للمتعارفين عليها، أو إذا تكونوا من أكثر من كلمة واحدة على سبيل المثال:

LINE.part.romeo { voice-family: Bruce, ؛Good Newsس, male }

خاصية الخطوة

تحدد خاصية pitch التردد الذي يستخدمه مؤلف المحادثة لنوع محدد من الكائنات. ولدرجة معينة، تتحكم هذه في إذا ما كان الصوت يشبه صوت الرجل أو المرآة. بينما، من الأفضل استخدام أصل. صوت مناسب بدلاً من ذلك تعطي القيمة بالهيرتز "دورة في كل ثانية" تكون أصوات النساء تقريباً 120Hz، بينما تكون الأصوات المعتادة للرجال في نطاق 200Hz.

يمكنك أيضاً استخدام تلك الكلمات الأساسية لتعدل الخطوة:

١- x-low

٢- low

٣- medium

٤- high

٥- x-high

تعد الترددات المضبوطة لتلك الكلمات الأساسية مستندة على بيئة المستخدم والصوت المحدد بينما يعد x-low دائماً أقل من low، والتي تعد دائماً أقل من medium وكذلك:

خاصية لنطاق الخطوة

تحدد خاصية pitch-range التنوع المقبول في متوسط خطوة المتحدث برقم يقع بين 0 و100. يتحكم في تغيير وتنوع الصوت المستخدم بواسطة مؤلف المحادثة تنشئ قيمة 0 صوت مستوى أحادي النغمة بينما يعد 50 صوت عادي، وتنشئ القيم فوق 50 صوت متحرك استثنائي.

خاصية Stress

تحدد خاصية Stress مستوى من التأكيد والتوضيح المستخدم في الصوت المتحدث. يكون الافتراضي 50. تعتبر لقيمة وتأثير تلك السمة تأثير مختلف في كل لغة يتم التحدث بها عند استخدامها مع لغات مثل الإنجليزية التي تؤكد على موضع الجملة، تستطيع أن تحدد نقاط الضغط الأولى، الثانية والثالثة للتحكم في التغيير الذي تم تطبيقه على مناطق الجملة.

خاصية Richness

تحدد خاصية Richness وضوح الصوت المستخدم بواسطة مؤلف المحادثة. كلما كان الصـــوت منسق، كلما كانت قدرة جملة أفضل. لا يتم حمل الأصوات الرفيقة بعيداً لأن أشكال موجاتـــها لا تكون بنفس عمق الخطوة مثل الصوت المنسق "الغني" أصوات تحمل أفضل بينمـــا تنتـــج القيـــم المنخفضة أصوات رفيعة "ناعمة" يسهل الاستماع إليها.

خصائص المحادثة

تتحكم تلك الخصائص في كيفية ترجمة مؤلف المحادثة للترقيم والأرقام. يوجد خاصيتين. خاصية speak-punctuation وخاصية speak-numeral.

خاصية Speak Punctuation

يتحدث الترقيم الافتراضي حرفياً. لعبارة مثل "The cat, Charm, ate all of his food." is read as "The cat comma Charm comma ate all of his food period" بينما بـإعداد خاصية speak-punctuation إلى none، لن يتم نطق أي الترقيم. ولكن سيكون لها إيقافـــات مؤقتة مثل أي صوت طبيعي متكلم علـــى ســـبيل المثـــال، "The cat <pause> Charm <pause> ate all of his food <silence>".

خاصية Speak Numeral

عند نطق الأرقام الافتراضية على شكل سلسلة مكتملة على سبيل المثال، يتم قراءة الرقـــم ١٠٢ "one hundred and two" بينما، إذا قمت بتعيين خاصية "one zero two " يمكنك أن تعـــود إلى الافتراض عن طريق إعداد خاصية speak-numeral إلى none، لن يتم نطق أي رقم.

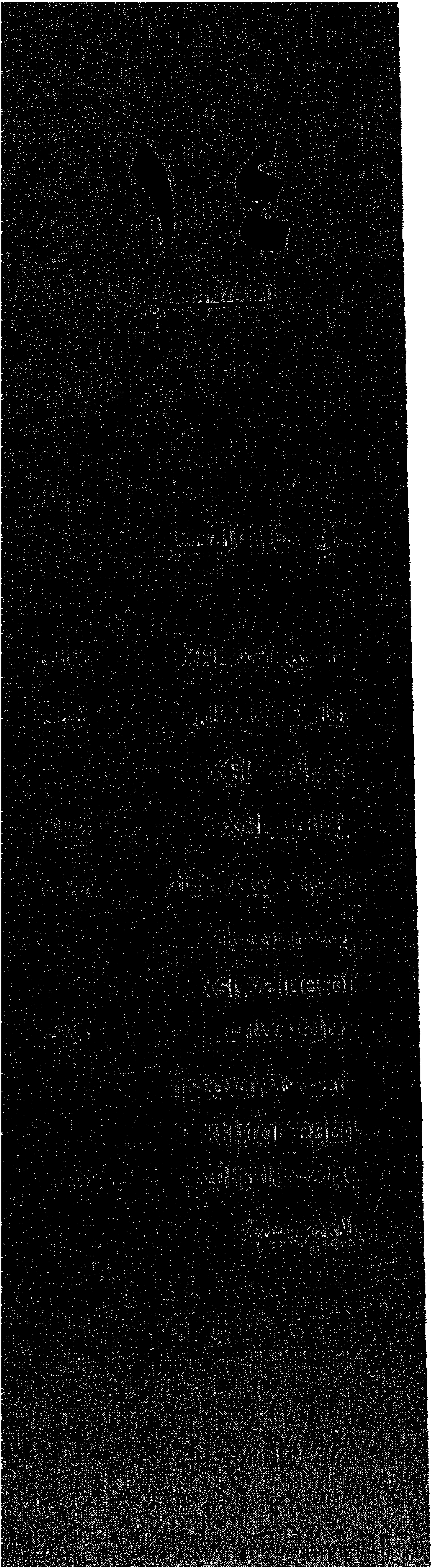
خلاصة

لقد قام هذا الفصل بتغطية مميزات CSS2 وكيفية استخدامها. ففي هذا الفصل تعلمت:

- يعد CSS2 غالباً مجموعة متطورة عن CSS1، ولهذا توجد اختلافات بسيطة بما في ذلك نوع معروض افتراضي instead بدلاً من block.
- مكتشف الإنترنت 5 وMozilla الهامش فقط الذي ينفذه CSS2 لذلك لا تتوقع أن تعمـــل الكثير من مميزاته في خلل.

- قام CSS2 بتوسيع المحددين المتنوعين الذين يمكنهم تطبيق خصائص محددة إلى عناصر معينة تتضمن محدد كوني، محددي التابع ومحددي الانحدار، ومحددي الارتباط.
- لقد تم تطوير قواعد @ الجديدة لمنح كاتبي المستند تحكم أكثر في مستنداتهم المطبوعة، بما في ذلك @charset و@page و@font-face.
- للـ CSS2 سبع فئات مُقترحة بما في ذلك: first-child: وhover: لتحديد العناصر التي لها شئ مشترك ولكن ليس من الضرورة أن يكون لها نفس النوع.
- للـ CSS2 اثنين من العناصر المُقترحة التي تمكنك من إدراج المحتوى في المستند :after و:before.
- للـ CSS2 لقد عملت CSS2 على زيادة استخدام خاصية العرض عن طريق دمج القيم لعرض العناصر مثل أجزاء الجدول كلها، وليس على الإطلاق "none"، ومثل كائنات compact أو run-in.
- تمكنك نظام الألوان وأنظمة الخطوط من إنشاء واجهة على تطبيقات XML الخاصة بك التي تطابق أكثر إعدادات النظام الأساسي لأجهزة الكمبيوتر الخاصة بالزائرين.
- تضيف CSS2 خصائص سمعية لوصف المحادثة، حجم الصوت، الإيقاف المؤقت المساعدة مميزات الصوت ومواصفات صوت يتم قراءته ومعرفة المكان الذي يأتي منه من بين الأشياء الأخرى.

مثل CSS1 لا يزال CSS2 حدود عديدة، أكثرها وضوحاً هي نقص الدعم الكامل من مستعرضي الويب، ولكن يجب أن يتغير هذا مع الوقت. لا يزال XSL أكثر لغو زرقة نمط في الفصل التالي، ستكتشف مع مستندات XML حتى الآن في الفصل التالي، ستكتشف تحويلات XSL، وسترى إلى أي مدى تستطيع تنفيذه.



لغة تحويل XSL

تتضمن لغة XSL وهي اختصار كلمة Extensible Style Language لغتين هما لغة تحويل ولغة تنسيق وكلاهما تطبيق للغة XML وتقدم لغة التمويل عناصر لتفريق القواعد التي يجب اتباعها لتمويل مستند XML إلى المستند الأصلي أو استخدام مجموعة مختلفة من العلامات ويفضل استخدام العلامات المعرفة بواسطة الجزء الثاني من XML وهو كائنات التنسيق ويعطي هذا الفصل نصف لغة XSL وهي لغة التمويل.

ما هي لغة XSL

يعمل كلاً من جزئي لغة XSL، جزء التمويل وجزء التنسيق، لصورة مستقلة عن بعضهم البعض فمثلاً يمكن تحويل مستند XML إلى ملف HTML صحيح التكوين باستخدام لغة التمويل وتجاهل كائنات تنسيق XSL تماماً وهذا هو نمط XSL المدعم بواسطة Internet Explore الذي تم عرضه في الفصل ٥ ويتم توضيحه بطريقة مفصلة في هذا الفصل.

بالإضافة إلى ذلك، لاستعلام إنتاج مستند تم كتابته باستخدام كائنات تنسيق XSL استخدام لغة تحويل XSL أخر فمثلاً يسهل تخيل محول تم كتابته لغة Java ويستطيع هذا المحول قراءة ملفات Tex وPDF ويتم تحويلهم إلى كائنات تنسيق XSL "هذا على الرغم من عدم صحة هذا التخيل إلى الآن ضيق ١٩٩٩".

تعتبر لغة XSL في حقيقة الأمر لغتين وليس لغة واحدة وهما لغة تحويل ولغة تنسيق ولكل اللغتين فائدته الخاصة دون الاستعانة باللغة الأخرى وتجعل لغة التمويل في تحريك البيانات من مستند XML إلى أخر مكون منهم للتجارة الإلكترونية التي تعتمد على XML مثل تناقل البيانات الإلكتروني وتبادل البيانات التفصيلية أو أي تطبيق أخر يتطلب التمويل بين مستندات XML المختلفة التي تحتوي على نفس البيانات وتتخذ تلك الاستخدامات في عدم اهتمامات بتقديم البيانات على شاشة العرض ليقرأها المستخدمين فهم يهتموا فقط بتحريك البيانات من برنامج أو نظام جهاز الكمبيوتر إلى أخر.

نتيجة لذلك ركزت العديد من تطبيقات XSL السابقة على لغة التمويل بصفة خاصة وتجاهلت كائنات التنسيق وعلى الرغم من كونها تطبيقات غير كاملة إلا أنها مفيدة فليست كل البيانات في حاجة إلى تقديمها على شاشة جهاز الكمبيوتر أو طباعتها على ورق.



يغطي الفصل ١٥ لغة تنسيق XSL.

تنويه عن XSL

يجري تطوير لغة XSL بصفة مستمرة فقد تغيرت لغة XSL بصورة أساسية في الماضي وسيستمر تغييرها في المستقبل ويعتمد هذا الفصل على مواصفات XSL المستخدمة في ٢١ أبريل ١٩٩٩ وعند وقت النسخة التمهيدية الرابعة قراءة هذا الكتاب قد تكون تلك النسخة التمهيدية قد تطورت وتم إدخال تغييرات على بناء جملة XSL الأساسية ولهذا إذا تعارض ما في هذا الكتاب مع النسخة المستخدمة من XSL يرجى مقارنة الأمثلة الموجودة في هذا الكتاب مع أحدث مواصفات لغة XSL.

لا يطبق أي برنامج حاليا كل مواصفات النسخة التمهيدية الرابعة ٢١ أبريل ١٩٩٩ ولا يطبقها أيضا جزء التمويل نطبق كل المنتجات المتاحة حاليا مجموعات فرعية من النسخة التمهيدية الحالية وبالإضافة إلى ذلك فالعديد من المنتجات ومن بينهم Internet Explorer 5.0 و XT تطبق عناصر ليست موجودة بالفعل في مواصفات النسخة التمهيدية الحالية الخاصة بلغة XSL لأن معظم المنتجات التي تحاول تطبيق ولو جزء من XSL تواجه مشكلات متعددة في الأجزاء التي تم تطبيق فيها XSL وبالتالي تعمل بعض الأمثلة البسيطة فقط بنفس الطريقة في برامج مختلفة.

وتحل تلك المشكلات كلما تقدم باتجاه الهدف النهائي حيث يصلح البائعين المشكلات التي تواجههم في المنتجات ويتم تطبيق النظام على الأجزاء التي لم يتم تطبيقه عليها بعد وتنشر العديد من البرامج التي تدعم XSL وإلى أن يتم ذلك فلك حرية اختيار أن العمل باستخدام XSL بصورته الحالية لغير الكاملة ومحاولة التعامل مع المشاكل التي تواجهه أو يمكن الالتزام بتقنية أكثر تطورا مثل CSS إلى أن تصبح XSL أكثر قوة.

نظرة عامة على تحويلات XSL

يقرأ معالج XSL في لغة تحويل XSL كلاً من مستند XML وورقة نمط XSL وبناء على التعليمات التي يجدها المعالج في ورقة نمط XSLتند XML جديد.

الشجرة

وضح الفصل ٦ أن كل مستند XML صحيح التكوين هو شجرة وتعتبر الشجرة بنية بيانات مكونة من "nodes" " متصلة تبدأ node" " فردية تسمى الجذور يتصل الجذر "nodes" التابعين له وقد تفضل كل منهم بصفر أو أكثر من العناصر التابعة تسمى "nodes" التي ليس لها عناصر تابعة أوراق ويبدو الرسم البياني للشجرة مثل تخطيط أصل النسب الذي يدرج أصل ملف واحد وأهم خاصية للشجرة أن كل nodes وعناصرها التابعة تكون شجرة وبناء على ذلك فإن الشجرة

هي ببنية هيكلية لأشجار حيث تبنى كل شجرة وبناء على ذلك فإن الشجرة هـــي ببنيـــة هيكليـــة لأشجار حيث تبنى كل شجرة من أشجار أصغر.

تعتبر nodes لشجرة XML هم العناصر ومحتوياتهم وعموماً يجب في XSL حساب السـمات وnamespaces وتعليمات المعالجة والتعليقات بالإضافة إلى ذلك يجب التميـيز بيـن للمسـتند والعنصر الجذري ولهذا تفترض معالجات XSL أن شجرة XML تحتوي على سبع أنـــواع مـن nodes وهم:

١- الجذر

٢- العناصر

٣- النص

٤- السمات

٥- Namespaces

٦- تعليمات المعالجة

٧- التعليقات

لننظر إلى مستند XML الموجود في تعليمات برمجة ١٤-١ ويوضح هذا المثال جدول دوري للعناصر التي تستخدم كمثال في هذا الفصل وهو بالتحديد يوضح أول عنصريـن فـي الجـدول الدوري.

يحتوي عنصر الجذر PERIODIC_TABLE على عناصر تابعـــة ATOM وكـل عنصـر ATOM يستضيف مجموعة متعددة من العناصر التابعة إلى تقدم الرقم الذري والـوزن الـذري والرموز ونقطة غليان العنصر وتحدد سمة UNITS عدد الوحدات للعناصر التي لها وحدات.



يبدو اختيار ELEMENT اختيار أفضل من ATOM إلا أن الكتابة عن عنـاصر ELEMENT وتنسيق محاولة التمييز أو بين العناصر الكيميائية وعنـاصر XML تعارض ولهذا السبب ففي هذا الفصل سيكون الاختيار هو ATOM.

تعليمات برمجة ١٤-١: الجدول الدوري في مستند XML وبه عنصرين هما الهيدروجين والهليوم

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="14-2.xsl"?>
<PERIODIC_TABLE>
```

```
<ATOM STATE="GAS">
  <NAME>Hydrogen</NAME>
  <SYMBOL>H</SYMBOL>
  <ATOMIC_NUMBER>1</ATOMIC_NUMBER>
  <ATOMIC_WEIGHT>1.00794</ATOMIC_WEIGHT>
  <BOILING_POINT UNITS="Kelvin">20.28</BOILING_POINT>
  <MELTING_POINT UNITS="Kelvin">13.81</MELTING_POINT>
  <DENSITY UNITS="grams/cubic centimeter"><!" At 300K ">
    0.0899
  </DENSITY>
</ATOM>

<ATOM STATE="GAS">
  <NAME>Helium</NAME>
  <SYMBOL>He</SYMBOL>
  <ATOMIC_NUMBER>2</ATOMIC_NUMBER>
  <ATOMIC_WEIGHT>4.0026</ATOMIC_WEIGHT>
  <BOILING_POINT UNITS="Kelvin">4.216</BOILING_POINT>
  <MELTING_POINT UNITS="kelvin">0.95</MELTING_POINT>
  <DENSITY UNITS="grams/cubic centimeter"><!" At 300K ">
    0.1785
  </DENSITY>
</ATOM>

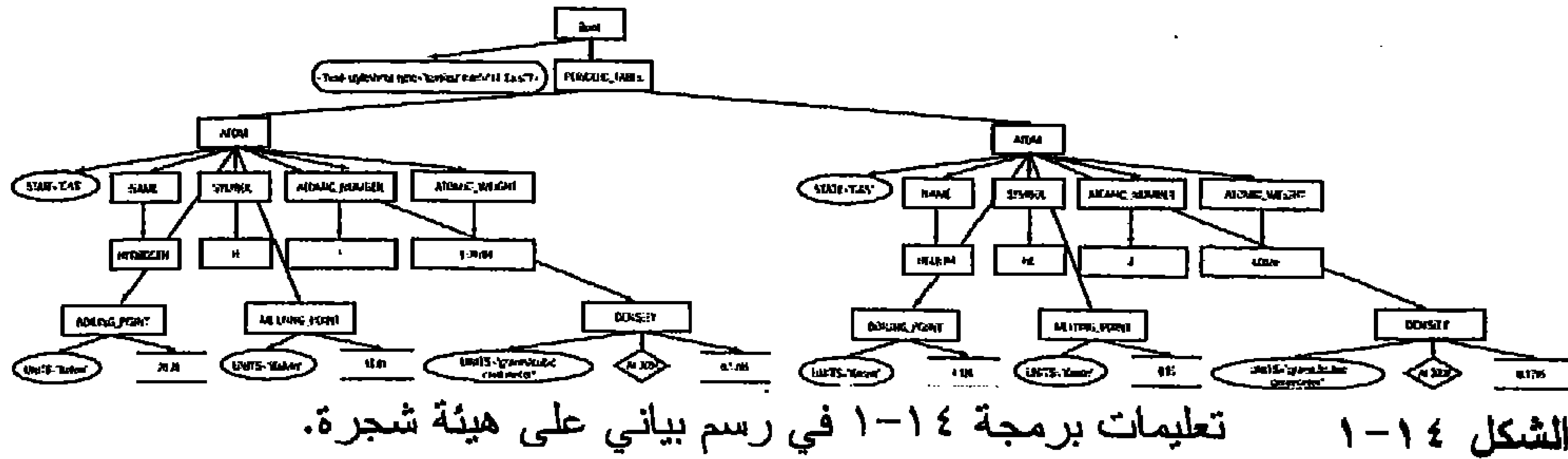
</PERIODIC_TABLE>
```

يعرض الشكل ١٤-١ رسم بياني لهذا المستند على هيئة شجرة وتبدأ تلك الشجرة في أعلاهـا بنقطة التقاء هي تحتوي على نقطتي التقاء تابعين تختلف عن العنصر الجـذري همـا تعليمـات معالجة xml-stylesheet والعنصر الجذري (The XML .PERIODIC_TABLE ويجب الانتباه إلى أن تعريف XML غير مرئي بالنسبة لمعالج XSL ويتضمنه الشجرة التي يعمل معـالج XSL فيها يتضمن عنصر PERIODIC_TABLE نقطتي التقاء تابعين كلاهما عناصر ATOM ولكـل عنصر ATOM نقطة التقاء بنية STATE وعدة نقاط التقاء للعناصر التابعة ويحتوي كل عنصـر تابع على نقاط التقاء لمحتوياته وأي سمات أو تعليقات خاصة بالعنصر وتختلف نقاط الالتقاء عـن

العناصر وهناك نقاط التقاء للنص والسمات والتعليقات وتعليمات المعالجة وعلى عكس CSS1، فإن لغة XSL غير مفيدة للعمل فقط مع العناصر الكاملة فللغة XSL رؤية شاملة للمستند والتي تتيح للمستخدم وضع أنماط بناء على التعليقات والسمات وتعليمات المعالجة وغير ذلك.



مثل تعريف XML فإن المجموعة الفرعية DTD الداخلية أو تعريف DOCTYPE ليس جزء من الشجرة إلا أنه قد يكون لها تأثير إضافة سمات نقاط التقاء لبعض العناصر "بمعنى أنه تم إضافة نقاط التقاء تكون سمة للعنصر" وذلك باستخدام تعريفات <!ATTLIST> التي تستخدم FIXED# أو قيم السمات الافتراضية.



الشكل ١٤-١ تعليمات برمجة ١٤-١ في رسم بياني على هيئة شجرة.

تعمل لغة تحويل XSL عن طريق تحويل شجرة XML أخرى وتحتوي اللغة على عوامل تشغيل لتحديد نقاط التقاء معينة من الشجرة وإعادة ترتيب نقاط الالتقاء وإخراج نقاط الالتقاء وتذكر أن كل عوامل التشغيل لكلاً من الإدخال والمخرجات صممت للعمل على شجرة فهم ليس لغة تعبير عامة لتحويل البيانات المعتادة.

مستندات أوراق نمط XSL

تقبل لغة تحويل XSL كإدخالات شجرة ممثلة في مستند XML وتنتج كمخرج شجرة جديدة ممثلة أيضاً في مستند XML وبالتالي يسمى جزء التمويل في لغة XSL جزء بناء الشجرة ويجب أن تكون المدخلات والمخرجات مستندات XML ولا يمكن استخدام لغة XSL لتحويل تنسيقات أخرى بخلاف تنسيقات XML مثل PDF أو Tex أو مايكروسوفت وورد أو PostScript أو MIDI أو أي شيء أخر ولكن يمكن استخدام XSL لتمويل XML إلى تنسيق وسط مثل TeXML ثم استخدم برنامج إضافي غير تابع للغة XSL لتمويل التنسيق الوسط إلى التنسيق المطلوب وتعتبر كل من HTML وSGML حالات غامضة بعض الشيء لقربها من XML ويمكن استخدام XSL للتمويل من وإلى HTML وSGML التي تلبي قواعد صحة تكوين مستندات XML مع ذلك لا تستطيع لغة XSL التعامل مع تنوع مستندات HTML وSGML غير صحيحة التكوين التي تكون على معظم

أنظمة إنتاج المستندات ومواقع الويب وأهم ما يجب معرفته هو أن لغة تحويل XSL تعمل بين تحويلات XML إلى XML وليس مع شيء أخر.

يحتوي مستند XSL على قائمة بقواعد القوالب وقواعد أخرى ولكل قاعدة قالب نموذج يحدد أنواع الأشجار التي تطبق عليها قاعدة القالب ويكون لها قالب لوضع المخرجات عليه عندما يتطابق النموذجان فعندما يقوم معالج XSL بتنسيق مستند XML باستخدام ورقة نمط XSL فإنه يقوم بفحص شجرة مستند XML وبفحص كل شجرة فرعية في دورها ويقارن المعالج أثناء قراءة كل شجرة في مستند XML بنموذج قاعدة القالب في ورقة النمط وعندما يجد المعالج شجرة تطابق نموذج قاعدة القالب فإنه يخرج قالب القاعدة وغالباً ما يحتوي هذا القالب على بعض الترميز والبيانات الجديدة وكذلك بعض البيانات المسموحة من شجرة مستند XML الأصلي.

تستخدم XSL لغة XML لوصف تلك القواعد والقوالب والنماذج بمستند XSL هو عنصر xsl:stylesheet وكل قاعدة قالب هي عنصر xsl:template نموذج القاعدة هو قيمة سمة match الخاصة بعنصر xsl:template وقالب المخرجات هو محتوى عنصر xsl:template ويتم أداء كل التعليمات الموجودة في القالب مثل تحديد أجزاء من إدخالات الشجرة لتضمينها في شجرة المخرجات بواسطة واحد أو أكثر من عناصر ويتم تعريفهم بكتابة XSL قبل اسم العنصر وتكون العناصر التي لا تحتوي على :XSL قبل اسم العنصر من شجرة النتائج.



تكون كل العناصر التي تتبع تعليمات XSL جزء من xsl name-space ويناقش الفصل ١٨ Namespaces "" وإلى أنه يصل إلى فصل ١٨؟ فكل ما يجب معرفته هو أن أسماء كل عناصر XSL تبدأ برمز XSL.

توضح تعليمات برمجة "١٤-٢" ورقة نمط XSL بسيطة تحتوي على قاعدتين من قواعد القوالب، تطابق أول قاعدة قالب العنصر الجذري PERIODIC_TABLE وتستبدل هذا العنصر بعنصر html ويكون محتوى عنصر html هو نتاج تطبيق القوالب الأخرى في المستند على المحتوى عنصر PERIODIC_TABLE.

يطابق القالب الثاني عناصر ATOM ويستبدل كل عنصر ATOM في مستند الإدخالات بعنصر P في مستند المخرجات وتدرج قاعدة xsl:apply-templates النص المطابق لعنصر المصدر في مستند المخرجات ولهذا يكون محتوى عنصر P هو النص الموجود في عنصر ATOM المتوافق معه وما يلي يناقش بناء الجملة لهذه العناصر بمزيد من التوسع.

تعليمات البرمجة ١٤-٢: ورقة نمط XSL للجدول الدوري مع وجود قاعدتي قوالب

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="PERIODIC_TABLE">
    <html>
      <xsl:apply-templates/>
    </html>
  </xsl:template>

  <xsl:template match="ATOM">
    <P>
      <xsl:apply-templates/>
    </P>
  </xsl:template>

</xsl:stylesheet>
```

أين يحدث تحويل XML

هناك ثلاثة طرق أساسية لتحويل مستندات XML إلى تنسيقات أخرى مثل HTML، باستخدام ورقة نمط XSL وهم:

١- يقدم كل من مستند XML وورقة النمط المقترنة به للعميل وهو مستعرض ويب والذي يقوم بتمويل المستند كما يحدد ورقة النمط وتقدمه للمستخدم.

٢- يطبق الخادم ورقة النمط XSL على مستند XML لتمويلها إلى تنسيق أخر في أغلب الأحيان HTML ويرسل المستند المحول إلى العميل وهو مستعرض ويب.

٣- يقوم برنامج ثالث بتمويل مستند XML الأصلي إلى تنسيق أخر وفي معظم الأحيان يكون HTML قبل وضع المستند على الخادم ويتعامل كلاً من الخادم والعميل مع المستند بعد تحويله.

يستخدم كلاً من هذه الاتجاهات برنامج مختلف على الرغم من استخدامهم لنفس مستند XML ونفس ورقة نمط XSL يوضح الاتجاه الأول استخدام مستعرض ويب عادي ليرسل مستندات XML إلى Internet Explorer 5.0 والاتجاه الثاني يوضحه استخدام خادم صغير يتوافق مع خادم ويب ويستخدم ذلك الخادم IBM alphaWorks يطبق XML وكمثال للاتجاه الثالث يمكن استخدام برنامج سطر الأوامر XT لتحويل مستندات XML إلى مستندات HTML ثم وضع مستندات HTML على خادم ويب ومع ذلك وكل تلك الأمثلة تستخدم من الناحية النظرية نفس لغة XSL.

نركز في هذا الفصل على الاتجاه الثالث والسبب في ذلك هو أنه في وقت كتابة هذا الكتاب تقدم برامج التمويل المتخصصة مثل James Clark's XT أو IBM's LotusXSL التطبيق الكامل والدقيق لمواصفات XSL الحالية وهذا يوفر أقصى درجات التوافق مع مستعرضات ويب وخادم ويب القانوني بينما يتطلب الاتجاه الأول مستعرض أحدث من المستخدم بواسطة معظم المستخدمين ويتطلب الاتجاه الثاني برنامج خادم ويب خاص وعلى الرغم من ذلك ففي التطبيق العملي تتطلب خادم مختلف ليس بأهمية تتطلب عميل معين فتستطيع المستخدم تثبيت برنامج الخادم الخاص به ولكنه لا يستطيع الاعتماد على أن زائري موقعه يثبتون برنامج عميل محدد.

كيفية استخدام XT

يعتبر XT هو تطبيق وضع الحرف Java 1.1 وتحتاج لتشغيله إلى آلة ظاهرية متوافقة مثل Sun's Java Development Kit (JDK) أو Java Runtime Environment (JRE) أو Apple's Macintosh Runtime for Java 2.1 (MRJ), أو آلة مايكروسوفت الظاهرية ويجب أيضاً تثبيت SAX متوافق مع موزع XML مثل James Clarkطs XP ويعتبر هذا إحدى تطبيقات Java.



في وقت كتابة هذا الكتاب توجد XT في الموقع http://www.jclark.com/xml/xt.html وتوجد XP في الموقع http://www.jclark.com/xml/xp/index.html وتتغير URL هذه بمرور الوقت ولا يوجد هناك ما يضمن وجود XT عند قراءة هذا الكتاب ومع ذلك يستخدم هذا الفصل XT وتعمل الأمثلة مع أي معالج XSL إلى تطبيق جزء من بناء الشجرة الخاص بالمواصفات التمهيدية للغة XSL التي تعمل من ٢١ أبريل ١٩٩٩ وهناك احتمال أخر للعمل وهو IBM alphaWorks' LotusXSL متاح في http://www.alphaworks.ibm.com/tech/LotusXSL وهناك احتمال أن

تعمل أو لا تعمل الأمثلة مع أي برنامج يطبق نسخة مواصفات تمهيدية أحدث من XSL وسيكتب أي تحديث في موقع الويب الخاص بالناشر في http://metalab.unc.edu/xml/books/bible/.

فئة Java التي تحتوي على الأسلوب الأساسي الخاص بتطبيق XT هو com.jclark.xsl.sax.Driver وعلى افتراض أن متغيرات بيئة CLASSPATH الخاصة بلغة Java تتضمن xt.jar و sax.jar وكلاهما متضمن في توزيع XT ويمكن تشغيل XT عن طريق كتابة ما يلي في مطالبة shell أو في إطار DOS.

```
C:\> java
  -Dcom.jclark.xsl.sax.parser=com.jclark.xml.sax.CommentDriver
   com.jclark.xsl.sax.Driver 14-1.xml 14-2.xsl 14-3.html
```

يشغل هذا الخط مترجم Java ويعين متغيرات بيئة Java com.jclark.xsl.sax.parser إلى com.jclark.xml.sax.CommentDriver والتي تشير إلى الاسم المؤهل الكامل لفئة Java المستخدمة لتوزيع مستندات الإدخال ويجب أن تكون تلك الفئة في مسار الفئة لدى المستخدم وتم استخدام في هذا المثال موزع XP ولكن أي موزع متوافق مع SAX سيفي بالغرض يلي ذلك اسم فئة Java التي تحتوي على أسلوب main() الخاص ببرنامج XT وهو com.jclark.xsl.sax.Driver في النهاية هناك أسماء إدخالات مستند XML "14-1.xml" وإدخالات ورقة نمط XSL "14-2.xsl" وكذلك مخرجات ملف HTML "14-3.html" وإذا تم إلغاء الوسيطة الأخيرة يتم طباعة المستند المحول على الواجهة.



في حالة استخدام برامج الويندوز وتم تثبيت آلة Java الظاهرية الخاصة لمايكروسوفت يمكن استخدام النسخة التنفيذية المستقلة بدأتها من XT وتعتبر تلك أسهل في الاستخدام حيث تتضمن موزع XP ولا تتطلب التعامل مع متغيرات بيئة CLASSPATH ويمكن، باستخدام هذا البرنامج، وضع ملف xt.exe في المسار وكتابة C:\> xt 14-1.xml 14-2.xsl 14-3.html.

تحول تعليمات برمجة "١٤-١" إدخالات المستندات إلى الملفات HTML صحيحة التكوين كما يوضح في الفصل ٦ وعموماً يمكن التمويل من أي تطبيق XML إلى أي تطبيق أخر طالما أمكن كتابة ورقة نمط لتدعيم التمويل ومثال لذلك يمكن تخيل ورقة نمط تحول من مستندات XML إلى مستندات SVG.

```
% java
  -Dcom.jclark.xsl.sax.parser=com.jclark.xml.sax.CommentDriver
  com.jclark.xsl.sax.Driver pinktriangle.vml
  VmlToSVG.xsl -out pinktriangle.svg
```

تتصرف سطور أوامر معالجات SXL الأحرف بنفس الطريقة تقريباً على الرغم من اختــلاف خيارات ووسائط سطر الأوامر وقد يكونوا أسهل في الاستخدام في حالة عدم كتابتهم بلغــة Java حيث لن تكون هناك حاجة إلى تكوين CLASSPATH.

توضح تعليمات برمجة "١٤-٣" مخرجات تشغيل تعليمات برمجة ١٤-١ مــن خــلال XT باستخدام ورقة نمط XSL الموجودة في تعليمات برمجة ١٤-٢ ولاحظ أن XT لا يحاول اختصار HTML التي يتم إنتاجها والتي تحتوي على مسافات بيضاء كثيرة ولا يعتبر ذلك شيئاً هاماً حيــث المراد هو عرض الملف في مستعرض ويب الذي يقوم بتقليل المسافات البيضاء ويوضح الشــكل ١٤-٢ تعليمات برمجة ١٤-٣ وقد تم تحميلها إلـــى Netscape Navigator 4.5 حيــث أن تعليمات برمجة ١٤-٣ تعرض HTML القياسي فلا تحتاج إلى مستعرض يستطيع عرض ملفــات XML لرؤيتها.

تعليمات البرمجة ١٤-٣: HTML منتجة بعد تطبيق ورقة النمط الموجودة في تعليمات برمجة ١٤-٢ على مستند XML في تعليمات برمجة ١٤-١

```
<html>

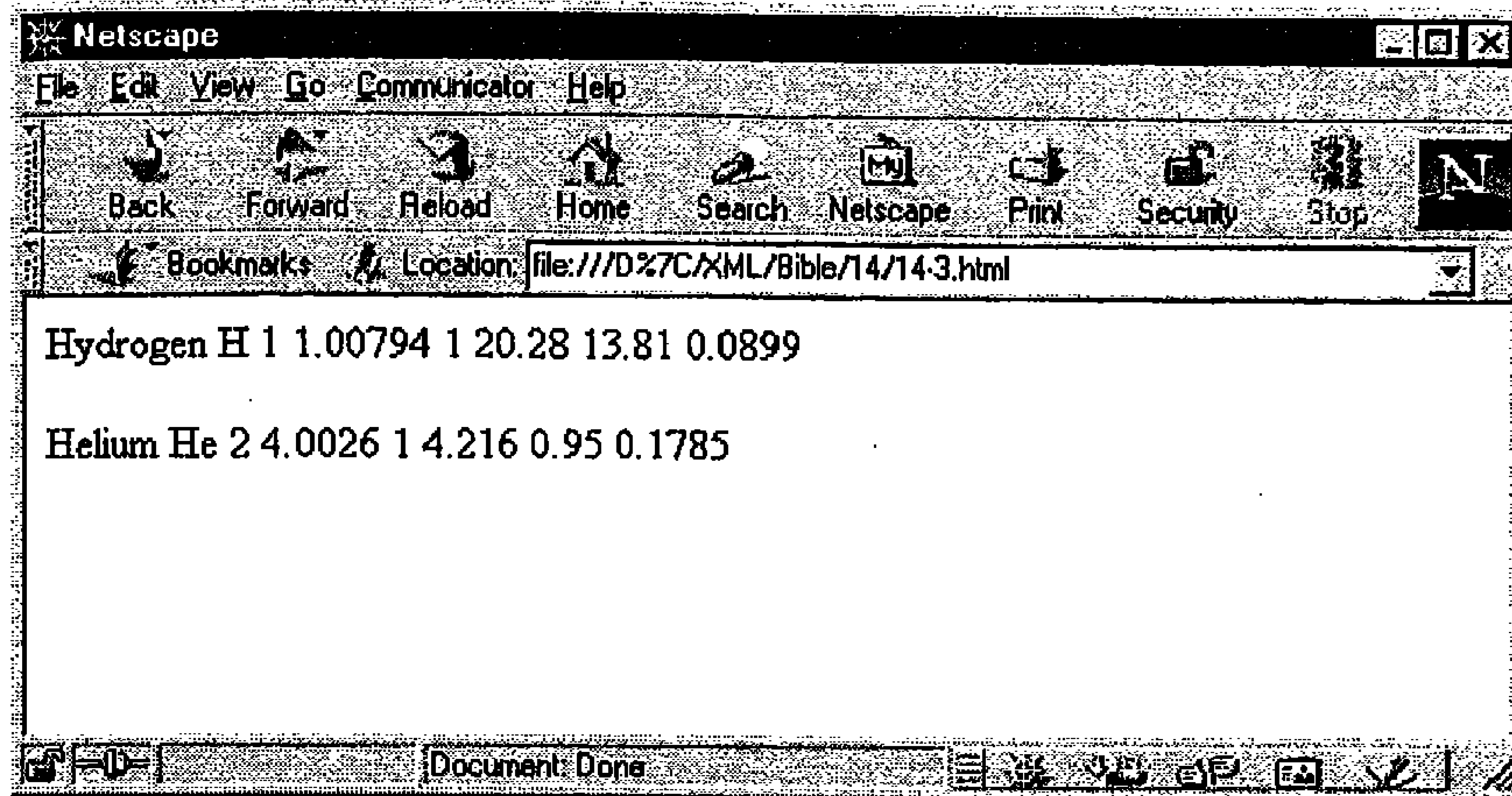
 <P>
  Hydrogen
  H
  1
  1.00794
  20.28
  13.81

   0.0899

 </P>

 <P>
  Helium
  He
  2
  4.0026
  4.216
  0.95
```

```
    0.1785
  </P>
</html>
```



الشكل ١٤-٢ الصفحة التي تم إنتاجها بعد تطبيق ورقة نمط الموجودة في تعليمات برمجة ١٤-٢ على مستند XML في تعليمات برمجة ١٤-١.

العرض المباشر للملفات XML مع أوراق نمط XSL

يمكن بدلاً من المعالجة المتبعة لملف XML إرساله إلى العميل بالإضافة إلى ملف XSL الذي يصف كيفية أدائه ويكون العميل مسئولاً عن تطبيق ورقة النمط على المستند وتشغيلها طبقاً لذلك وعلى الرغم من أن ذلك يعتبر تحميل عمل زائد على العميل إلا أنه يقلل الحمل عن الخادم وفي هذه الحالة يجب أن تحول ورقة نمط XSL المستند على تطبيق XML يفهمه العميل وتعتبر لغة HTML اختيار مناسب على الرغم من بعض المستعرضات في المستقبل ستعمل مع كائنات تنسيق XSL أيضاً.

يعتبر إلحاق ورقة نمط XSL إلى مستند XML عملاً سهلاً. أدرج تعليمات معالجة xml-stylesheet في تسجيل الدخول التمهيدي مباشرة بعد تعريف XML ويجب أن تحتوي تعليمات المعالجة تلك على سمة type مع قيمة text/xsl وسمة href التي تعتبر قيمتها URL موجهة إلى ورقة النمط ومثال لذلك.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="14 - 2.xsl"?>
```

وهذه هي طريقة إلحاق ورقة نمط CSS إلى المستند والفرق الوحيد هو أن سمة type لها قيمة text/xsl بدلاً من text/css.

يختلف دعم Internet Explorer 5.0's للغة XSL عن المواصفات التمهيدية العاملة من تاريخ ٢١ أبريل ١٩٩٩ في عدة أوجه أولها أنه يتوقع أن تكون عناصر XSL موجودة في http://www.w3.org/TR/WD-xsl "namespace" بدلاً من http://www.w3.org/XSL/ Transform/1.0 "namespace" على الرغم من أن استخدام XSL البادئة ثانياً أنها تطبق القواعد الافتراضية للعناصر التي لا تطابق القوالب وبالتالي تحتاج إلى تزويد قالب لكل عنصر في الهيكل بدءاً من الجذر قبل محاولة عرض المستند في Internet Explorer التي توضحها تعليمات برمجة "١٤-٤" تطابق القواعد الثلاثة نقطة بدء الجذر وعنصر الجذر PERIODIC_TABLE وعنصر ATOM بنفس الترتيب ويوضح شكل ١٤-٣ مستند XML في تعليمات برمجة ١٤-١ وقد تم تحميله إلى Internet Explorer 5.0 مع ورقة النمط.

تعليمات البرمجة ١٤-٤: ضبط ورقة نمط تعليمات برمجة ١٤-٢ للعمل مع Internet Explorer 5.0

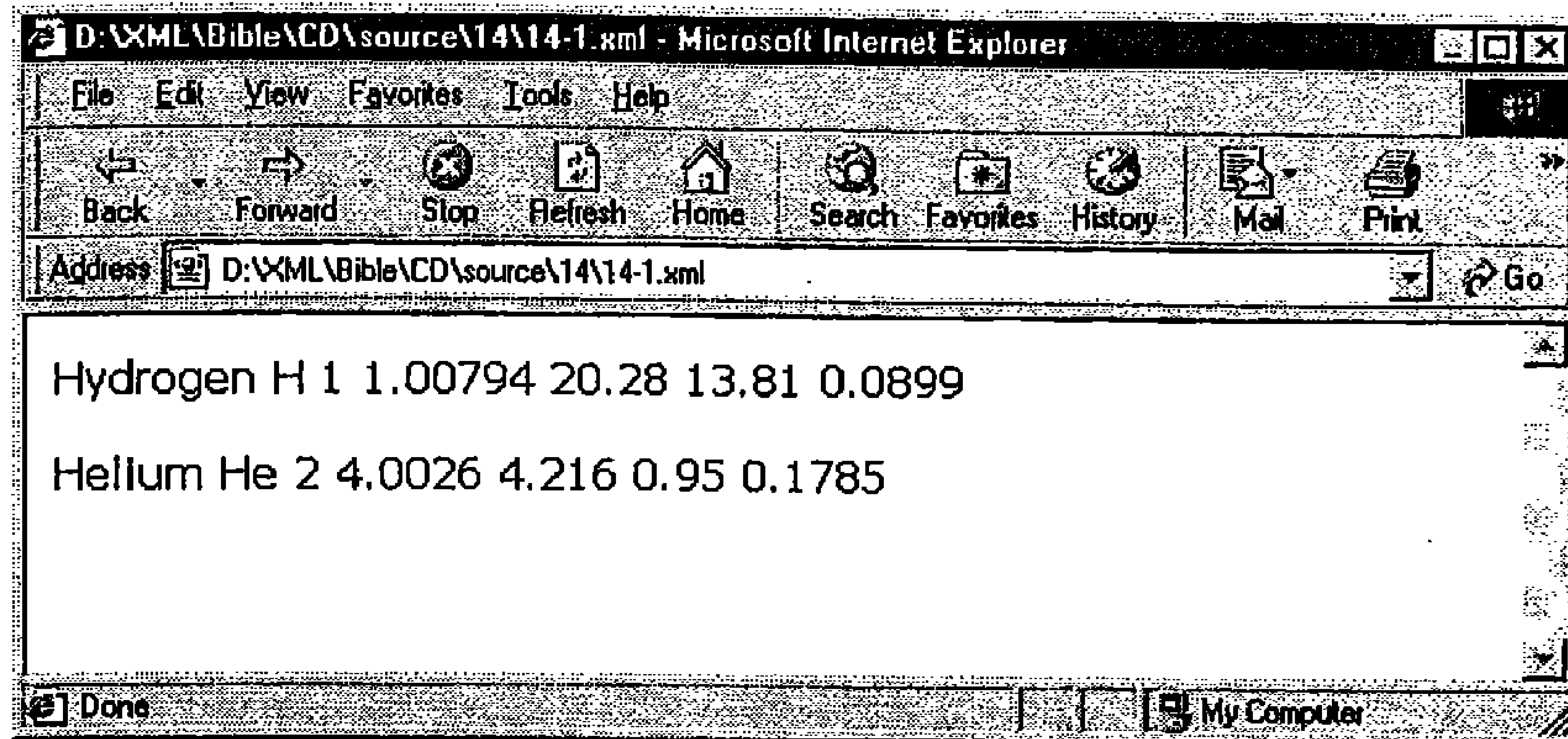
```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

  <xsl:template match="/">
    <html>
      <xsl:apply-templates/>
    </html>
  </xsl:template>

  <xsl:template match="PERIODIC_TABLE">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="ATOM">
    <P>
      <xsl:value-of select="."/>
    </P>
  </xsl:template>
```

```
</xsl:stylesheet>
```



الشكل ١٤-٣ الصفحة التي استخدم Internet Explorer 5.0 لإنتاجـــها بتطبيق ورقة نمط XSL التي تم ضبطها في تعليمات برمجـــة ١٤-٤ على مستند XML في تعليمات برمجة ١٤-١

تنبيه: يمكن استخدام نفس مستند XML لكل من العرض المباشر والتقديم المســبق إلـــى HTML ولكن للأســـف لا تقبــل http://www.w3.org/ TR/WD-xsl XT namespace ولن تقبـــل http://www.w3.org/XSL/ Transform/1.0 namespace والسبب في تلك هو تقدم المعالجات المختلفة على بعضها البعــض في دعم الإجراء المختلفة من مواصفات XSL المتطورة.

في بقية الفصل يتم التقديم المسبق للملف في HTML قبل تحميله إلى مستعرض ويب.

قوالب XSL

تعتبر قواعد القوالب التي تم تعريفها بواسطة xsl:template هي أهم جزءمن ورقة نمــط XSL وكل قاعدة قالب هي عنصر xsl:template وربط قواعد القوالب هذه بين مخرجـــات محـــددة وإدخالات محددة ولكل عنصر xsl:template سمة match التي تحدد أي نقـــاط بـــدء مســـتند الإدخال يكون القالب مرتبط بها instantiated.

القالب الأساسي الذي يتم الارتباط هو محتوى عنصر xsl:template وقد يحتوي القالب على كلاً من النص الذي سيظهر حرفياً في مستند المخرجات وتعليمات XSL التي تنسخ البيانات مـــن مستند إدخال XML إلى النتائج ولأن جميع تعليمات XSL توجد في xsl namespace أي أن تبدأ :xsl فمن السهل التمييز بين العناصر التي تعتبر بيانات حرفية يتـــم نســـخها إلـــى المخرجـــات

وتعليمات XSL ومثال لذلك ما يلي هو قالب تم تطبيقه على نقطة بدء الجزء في شجرة الإدخال.

```
<xsl:template match="/">
  <html>
    <head>
    </head>
    <body>
    </body>
  </html>
</xsl:template>
```

عندما يقوم معالج XSL بقراءة مستند الإدخال فإن أول نقطة بدء يراها هي الجذر وتطابق هذه القاعدة نقطة بدء الجزء وتعلم معالج XSL أن يعبر عن هذا النص بما يلي:

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

يعتبر هذا النص لغة HTML صحيحة التكوين وحيث أن مستند XSL نفسه هو مستند XML فإن محتوياته بما في ذلك القوالب يجب أن تكون مستند XML صحيح التكوين.

في حالة استخدام القاعدة السابقة فقط في ورقة نمط XSL فإن المخرجات تكون محددة في المعلومات الست السابقة وفي الواقع قد تم حفظهم إلى الأربع علامات المساوية وهم: <html><head/><body/></html> وهذا لأنه لا يوجد إرشادات في القاعدة تطلع المنسق على ضرورة التحرك باتجاه الأسفل في الشجرة والبحث عن أشياء أخرى تطابق القوالب الموجودة في ورقة النمط.

عنصر xsl:apply-templates

للانتقال إلى ما بعد الحذر يجب أن يقوم محرك التنسيق لمعالجة العناصر التابعة للجذر وعموماً ليتم تضمين المحتوى في نقاط البدء التابعة يجب معالجة نقاط البدء بصورة متصلة من خلال مستند XML والعنصر الذي يقوم بتنفيذ ذلك هو عنصر xsl:apply-templates يعلم المنسق أنه بتضمين عنصر xsl:apply-templates في قالب المخرجات يجب عليه مقارن كل عنصر تابع لعنصر المصدر المطابق مع القوالب في ورقة النمط وفي حالة وجود تطابق يتم إخراج القالب

الخاص بنقطة البدء المطابقة وقد يحتوي القالب الخاص بنقطة البدء المطابقة على عناصر xsl:apply templates للبحث عن ما يطابق العناصر التابعة الخاصة به وعندما يقوم محرك التنسيق لمعالجة نقطة بدء يتم معاملة نقطة البدء على أساس أنها شجرة كاملة وهذا من مميزات بناء الشجرة حيث يمكن معاملة كل جزء على أنه الكل ومثال لذلك تعليمات برمجة ١٤-٥ وهي عبارة عن ورقة نمط XSL تستخدم عنصر xsl:apply templates لمعالجة نقاط البدء التابعة لها.

تعليمات البرمجة ١٤-٥: ورقة نمط XSL التي تقوم بصورة متصلة بمعالجة العناصر التابعة للجذر

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/">
    <html>
<xsl:apply-templates/>
    </html>
  </xsl:template>

  <xsl:template match="PERIODIC_TABLE">
    <body>
      <xsl:apply-templates/>
    </body>
  </xsl:template>

  <xsl:template match="ATOM">
    An Atom
  </xsl:template>

</xsl:stylesheet>
```

ما يلي هو ما يحدث عند تطبيق ورقة النمط السابقة على تعليمات برمجة ١٤-١:

١- تتم مقارنة نقطة بدء الجذر مع كل قواعد القوالب في ورقة النمط وتظهر المطابقة مع

أول قاعدة.

٢- تكتب علامة <html>.

٣- يتسبب عنصر xsl:apply-templates في جعل محرك التنسيق يعالج نقاط بدء العناصر التابعة.

أ- تتم مقارنة أول عنصر تابع للجذر وهو تعليمات معالجة xsl:apply مع قواعد القوالب وحيث أنه لا توجد مطابقة فلا يوجد مخرجات.

ب- تتم مقارنة ثاني العناصر التابعة للجذر وهو عنصر الجذر PERIODIC_TABLE مع قواعد القوالب والنتيجة أنها تطابق قاعدة القالب الثانية.

ج- تكتب علامة <body>.

د- يتسبب وجود عنصر xsl:apply-templates في عنصر body في جعل محرك التنسيق يقوم بمعالجة نقاط بدء العناصر التابعة الخاصة بالعنصر PERIODIC_TABLE.

د.١- تتم مقارنة أول عنصر تابع لعنصر PERIODIC_TABLE وهو عنصر الهيدروجين ATOM مع قواعد القوالب وهو يطابق قاعدة القالب الثالث.

د.٢- يصبح النص "An Atom" أنه ذرة مخرج.

د.٣- تتم مقارنة العنصر التابع الثاني لعنصر PERIODIC_TABLE وهو عنصر الهليوم ATOM مع قواعد القوالب ويطابق قاعدة القالب الثالثة.

د.٤- يصبح النص "An Atom" أنه ذرة مخرج.

د.٥- تكتب علامة </body>.

٤-تكتب علامة </html>.

٥-انتهت عملية المعالجة.

النتيجة النهائية هي:

```
<html><body>
An Atom
  An Atom
</body></html>
```

سمة select

لاستبدال النص "An Atom" باسم عنصر ATOM كما يوضح العنصر التـــابع NAME ويجـــب تحديد أن تلك القوالب تطبق على العناصر التابعة NAME الخاصة بعنصـــر ATOM ولاختيـــار مجموعة محددة من العناصر التابعة بدلاً من كل العنـــاصر التابعـــة يجـــب إمـــداد xsl:apply-templates بسمة select تعيين العناصر التابعة المراد تحديدها وفي هذا المثال هي:

```
<xsl:template match="ATOM">
  <xsl:apply-templates select="NAME"/>
</xsl:template>
```

تستخدم سمة select نفس نماذج سمة match لعنصر xsl:template element وفي تلـــك المرحلة ستلتزم بأسماء العناصر البسيطة ولكن في الجزء الخاص بالنماذج المستخدمة للمطابقـــة والتحديد فيما بعد في هذا الفصل ستقوم باستكشاف إمكانيات أكثر لكل من سمة select وفي حالة عدم تحديد سمة select يتم تحديد كل العناصر التابعة.

يوضح ما يلي نتيجة إضافة هذه القاعدة إلى ورقة نمط تعليمات برمجة ١٤-٥" وتطبيقها علـــى تعليمات برمجة ١٤-٥.

```
<html><head/><body>

 Hydrogen

 Helium

</body></html>
```

تحديد قيمة نقطة بدء باستخدام xsl:value-of

ينسخ عنصر xsl:value-of قيمة نقطة بدء في مستند الإدخال إلى مستند المخرجات وتحدد سمة select لعنصر xsl:value أي قيمة نقطة بدء يتم أخذها.

لنفترض مثلاً أن المستخدم يرغب في استبدال النص الحرفـــي "An Atom" باســـم عنصـــر ATOM كما توضح محتويات العنصر التابع له NAME ويمكن إبـــدال An Atom كمـــا يلـــي "<xsl:value-of select="NAME"/" مثل ما يلي:

```
<xsl:template match="ATOM">
  <xsl:value-of select="NAME"/>
</xsl:template>
```

ثم بعد ذلك عند تطبيق ورقة النمط على تعليمات برمجة ١٤-١ يكون النص الناتج كما يلي:

```
<html><head/><body>

 Hydrogen

 Helium

</body></html>
```

يرتبط العنصر الذي يتم تحديد قيمته وفي هذا المثال هو عنصر NAME يصله بنقطة بدء المصدر وتعتبر نقطة بدء المصدر هي العنصر الذي تتم مطابقته بواسطة القالب وفي هذا المثال هو عنصر ATOM المحدد ولذا عندما تتم مطابقة عنصر ATOM لغاز هيدروجين بواسطة <xsl:template match="ATOM"> ويتم تحديد NAME الخاصة بعنصر ATOM لغاز الهيدروجين بواسطة xsl:value وعندما تتم مطابقة ATOM الخاصة بغاز الهليوم بواسطة <xsl:template match= "ATOM"> فيتم تحديد NAME لعنصر ATOM الخاص بغاز الهليوم بواسطة xsl:value-of.

تكون قيمة نقطة البدء سلسلة في أغلب الأحيان ومن الممكن أن تكون سلسلة فارغة وتعتقد المحتويات الفعلية لهذه السلسلة على نوع نقطة البدء وأكثر أنواع نقاط البدء شيوعاً هي العنصر وقيمة نقطة بدء العنصر بسيطة بالفعل وهي سلسلة متصلة من كافة بيانات الأحرف الموزعة بين علامة بدء العنصر وعلامة النهاية. وهي ليست ترميز ومثال لذلك عنصر ATOM الأول في تعليمات برمجة ١٤-١ هو كما يلي:

```
<ATOM STATE="GAS">
 <NAME>Hydrogen</NAME>
 <SYMBOL>H</SYMBOL>
 <ATOMIC_NUMBER>1</ATOMIC_NUMBER>
 <ATOMIC_WEIGHT>1.00794</ATOMIC_WEIGHT>
 <OXIDATION_STATES>1</OXIDATION_STATES>
 <BOILING_POINT UNITS="Kelvin">20.28</BOILING_POINT>
 <MELTING_POINT UNITS="Kelvin"13.81</MELTING_POINT>
 <DENSITY UNITS="grams/cubic centimeter"><!" At 300K ">
  0.0899
 </DENSITY>
</ATOM>
```

ما يلي هو قيمة هذا العنصر:

Hydrogen
H
1
1.00794
1
20.28
13.81

0.0899

تم حساب ذلك عن طريق استبعاد كل العلامات والتعليقات أما بالنسبة لما تبقى بما في ذلك المسافات البيضاء لم تقترب منهم ويتم حساب قيم أنواع نقاط البدء الست الأخرى بنفس الأسلوب وهو أسلوب واضح ويلخص الجدول ١٤-١ ذلك.

الجدول ١٤-١
قيم نقاط البدء

نوع نقطة البدء	القيمة
الجذر	هو قيمة عنصر الجذر.
العنصر	هو تسلسل بيانات الأحرف الموزعة المحتواة في العنصر بما في ذلك بيانات الحرف في أي من جيل العنصر.
النص	هو نص نقطة البدء أساساً هي نقطة البدء نفسها.
السمة	هي قيمة السمة المتساوية كما يتم تحديدها بواسطة Section 3.3.3 لتوصيات XML 1.0 وأساساً قيمة السمة من حل entities ويتم تخطيط المسافات البيضاء البادئة واللاحقة وهي لا تتضمن اسم السمة وعلامات المساواة أو علامات الاقتباس.
Namespace	هي URI الخاصة namespace.
تعليمات المعالجة	قيمة تعليمات المعالجة لا تتضمن اسم تعليمات المعالجة وهي <? or ?>.
التعليق	لا يتم تضمين نص التعليق <!-- and -->

معالجة عناصر متعددة باستخدام xsl:for-each

يستخدم عنصر xsl:for-each فقط في سياق لا يكون العنصر غامض فيها بالنسبة لقيمة نقطة البدء التي يتم تناولها، فإذا كان هناك عدة عناصر محتملة يمكن تحديدها قي اختيار العنصر الأول فقط فمثلاً هذه قاعدة ضعيفة لأن عنصر PERIODIC_TABLE الاعتيادي يحتوي على أكثر من ATOM.

```
<xsl:template match="PERIODIC_TABLE">
  <xsl:value-of select="ATOM"/>
</xsl:template>
```

هناك طريقتين لمعالجة العناصر المتعددة على التوالي وأول تلك الأساليب هي استخدام xsl:apply-templates مع سمة select التي تختار العناصر المحددة المراد تضمينها مثل ما يلي:

```
<xsl:template match="PERIODIC_TABLE">
   <xsl:apply-templates select="ATOM"/>
 </xsl:template>

 <xsl:template match="ATOM">
   <xsl:value-of select="."/>
 </xsl:template>
```

يحدد عنصر "."=select في القالب الثاني للمنسق أن يأخذ قيمة العنصر المنطلق وفي هذا المثال هي ATOM.

الخيار الثاني هو xsl:for-each ويقوم عنصر xsl:for-each لمعالجة كل عنصر يتم اختياره بواسطة سمة select بالتوالي ومع ذلك لا يتطلب ذلك أي قوالب إضافية ومثال ذلك:

```
<xsl:template match="PERIODIC_TABLE">
  <xsl:for-each select="ATOM">
    <xsl:value-of select="."/>
  </xsl:for-each>
</xsl:template>
```

في حالة حذف سمة select يتم معالجة كل العناصر التابعة لنقطة بدء المصدر وهي في هذا المثال PERIODIC_TABLE.

```
<xsl:template match="PERIODIC_TABLE">
  <xsl:for-each>
```

```
    <xsl:value-of select="ATOM"/>
  </xsl:for-each>
</xsl:template>
```

نماذج مطابقة نقطة البدء

تدعم سمة match الخاصة بعنصر xsl:template بناء جملة معقدة يتيح التغيير بدقة عن نقاط البدء المراد مطابقتها ونقاط البدء التي لا يرغب المستخدم في مطابقتها وكذلك تدعم سمة select الخاصة بكل من العناصر التاليـــة: xsl:apply-templates و xsl:value-of و xsl:for-each و xsl:copy-of و xsl:sort مجموعة أعلى من بناء الجملة وأكثر قوة تتيح التعبير بالتحديد عـــن نقاط البدء المراد تحديدها ونقاط البدء التي لا يرغب المستخدم في تحديدها وفيمـــا يلـــي نمـــاذج متعددة لمطابقة وتحديد نقاط بدء.

مطابقة نقطة بدء الجذر

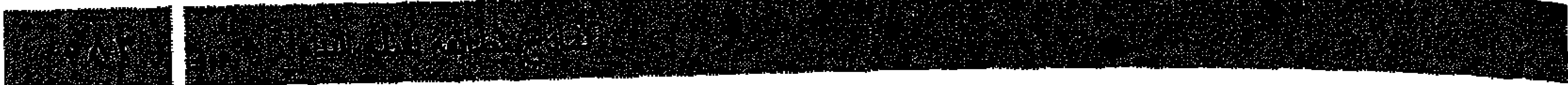
للحصول على مستند مخرجات صحيح التكوين يجب أن يكون أول المخرجات من تحويـــل XSL هو عنصر الجذر لمستند المخرجات وبالتالي غالباً ما تبدأ ورقة نمط XSL بقاعدة تطبـــق علـــى نقطة بدء الجذر ولتحديد نقطة بدء الجذر في قاعدة يجب إعطاء سمة match الخاصة بها قيمـــة "/" ومثال ذلك:

```
<xsl:template match="/">
  <html>
    <xsl:apply-templates/>
  </html>
</xsl:template>
```

تطبق هذه القاعدة على نقطة بدء الجذر الخاصة بشجرة الإدخال فقط عند قراءة نقطـــة بـــدء الجذر تصبح علامة <html> مخرج تتجاوز هذه القاعدة الافتراضية لنقطة بدء الجذر وتوضـــح تعليمات برجة ١٤-٦ ورقة بقاعدة فردية تطبق على نقطة بدء الجذر:

تعليمات البرمجة ١٤-٦: ورقة نمط XSL بقاعدة واحدة لنقطة بدء الجذر

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
```



```
<xsl:template match="/">
  <html>
    <head>
      <title>Atomic Number vs. Atomic Weight</title>
    </head>
    <body>
      <table>
        Atom data will go here
</table>
    </body>

  </html>
 </xsl:template>

</xsl:stylesheet>
```

تقدم ورقة النمط هذه قاعدة لنقطة بدء الجذر وحيث أن قالب هذه القاعدة لا يحدد أي معالجات أخرى لنقاط بدء العناصر التابعة وتكون المخرجات الحرفية وهي ما تراه على القالب هي كل مـا يتم إدراجه في مستند النتيجة وبمعنى أخر فيما يلي هو نتاج تطبيق ورقة النمط الموجـودة فـي تعليمات برمجة ١٤-٦على تعليمات برمجة ١٤-١ أو أي مستند XML صحيح التكوين:

```
<html><head><title>Atomic Number vs. Atomic
Weight</title></head><body><table>
          Atom data will go here
        </table></body></html>
```

مطابقة أسماء العناصر

كما ذكر سابقاً، يحتوي النموذج الأساسي اسم عنصر فردي يطابق كل العناصر التي تحمل نفـس الاسم ومثال لذلك يطابق هذا القالب عناصر ATOM ويضع علامة سميكة على العناصر التابعـة لعنصر ATOMIC_NUMBER.

```
<xsl:template match="ATOM">
  <b><xsl:value-of select="ATOMIC_NUMBER"/><b>
</xsl:template>
```

توضح تعليمات برمجة ١٤-٧ ورقة نمط تتسع عن تعليمات ١٤-٦ أولاً يتم تضمين عنصـر xsl: apply-templates في قالب قاعدة نقطة بدء الجذر وتستخدم تلك القـاعدة سـمة select لتأكيد أن عناصر PERIODIC_TABLE فقط يتم معالجتها.

ثانياً، يتم نشاء قاعدة تطبيق على عناصر PERIODIC_TABLE باستخدام match="PERIODIC_TABLE" وتعيين هذه القاعدة رأس الصفحة للجدول ثم تطبيق القوالب لتكوين جسم الجدول من عناصر ATOM.

في النهاية تحدد قاعدة ATOM عناصر ATOM التالية NAME وATOMIC_ NUMBER وATOMIC_WEIGHT وهي عناصر تابعة باستخدام <xsl:apply-templates select="NAME"/> و<xsl:apply-templates select="ATOMIC_NUMBER""/> و<xsl:apply-templates select="ATOMIC_WEIGHT"/> ويتم تجميع كل ذلك في داخل عناصر tr وtd في HTML لتكون النتيجة النهائية جدول الأرقام الذرية يتطابق مع الأوزان الذرية ويوضح شكل ١٤-٤ مخرجات تطبيق ورقة النمط في تعليمات برمجة ١٤-٧ لكل مستند الجدول الدوري كاملاً.

يجب ملاحظة ما يلي بشأن ورقة النمط وهو أن الترتيب الأساسي لعناصر NAME, ATOMIC_NUMBER وATOMIC_WEIGHT في إدخال المستند الذي له صلة بالمخرجات حيث يظهروا في المخرجات بالترتيب الذي يتم تحديدهم به وهو الرقم أولاً ثم الوزن وبالعكس يتم فرز الذرات الفردية وفقاً للترتيب الهجائي كما يظهروا في مستند الإدخال وفيما بعد يوضح كيفية استخدام عنصر xsl:sort لتغيير ذلك يمكن ترتيب الذرات بترتيب العدد الذري المناسب.

تعليمات البرمجة ١٤-٧: قوالب مطبقة على فئات محددة لعناصر تحتوي على عنصر select

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/">
   <html>
     <head>
       <title>Atomic Number vs. Atomic Weight</title>
     </head>
     <body>
       <xsl:apply-templates select="PERIODIC_TABLE"/>
     </body>
   </html>
  </xsl:template>

  <xsl:template match="PERIODIC_TABLE">
   <h1>Atomic Number vs. Atomic Weight</h1>
   <table>
     <th>Element</th>
```

```
    <th>Atomic Number</th>
    <th>Atomic Weight</th>
      <xsl:apply-templates select="ATOM"/>
  </table>
</xsl:template>

<xsl:template match="ATOM">
  <tr>
    <td><xsl:value-of select="NAME"/></td>
    <td><xsl:value-of select="ATOMIC_NUMBER"/></td>
    <td><xsl:value-of select="ATOMIC_WEIGHT"/></td>
  </tr>
</xsl:template>

</xsl:stylesheet>
```

Atomic Number vs. Atomic Weight - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: file:///D%7C/XML/Bible/14/14-8.html

Atomic Number vs. Atomic Weight

Element	Atomic Number	Atomic Weight
Actinium	89	227
Silver	47	107.868
Aluminum	13	26.98154
Americium	95	243
Argon	18	39.948
Arsenic	33	74.9216
Astatine	85	210
Gold	79	196.9665
Boron	5	10.811
Barium	56	137.33
Beryllium	4	9.01218
Bohrium	107	262
Bismuth	83	208.9804
Berkelium	97	247

Document: Done

الشكل ١٤-٤ جدول يوضح العدد الذري فـي مقـابل الـوزن الذري في Netscape Navigator 4.5.

مطابقة العناصر التابعة باستخدام علامة /

لا يتم تقييد المستخدم بالعناصر التابعة لنقطة البدء الحالية في سمة match ويمكن استخدام علامة (/) لمطابقة هياكل العناصر المحددة وفي حالة استخدام علامة (/) وحدها فإنها تشير إلى نقطة بدء الجذر ومع ذلك يمكن استخدامها بين اسمين للإشارة إلى أن الاسم الثاني هو العنصر التابع للاسم الأول ومثال لذلك تشير ATOM/NAME إلى عناصر NAME التي تكون عناصر تابعة لعناصر ATOM.

يتاح للمستخدم في عناصر xsl:template مطابقة بعض عناصر نوع معطى ومثال لذلك تمييز قاعدة القالب هذه عناصر SYMBOL التي تكون عناصر تابعة لعناصر ATOM المركبة وتلك القاعدة لا تؤثر على عناصر SYMBOL التي لا تكون عناصر تابعة مباشرة لعناصر ATOM.

```
<xsl:template match="ATOM/SYMBOL">
  <strong><xsl:value-of select="."/></strong>
</xsl:template>
```



تذكر أن هذه القاعدة تحدد عناصر SYMBOL التي تكون عناصر تابعة لعناصر ATOM وليس عناصر ATOM التي لها عناصر SYMBOL تابعة وبمعنى أخر فعلامة. "النقطة" في <xsl:value-of select="."/> تشير إلى SYMBOL وليس إلى ATOM.

يمكن تحديد مطابقات أعمق عن طريق وضع النماذج في سلاسل معاً فمثلاً سلسلة PERIODIC_TABLE/ATOM/NAME تحدد عناصر NAME الذي يكون أصلها عنصر ATOM ويكون أصله عنصر PERIODIC_TABLE.

تستخدم أيضاً علامة (*) البديلة لاستبدال أي اسم عنصر في هيكل ومثال لذلك تطبيق قاعدة القالب لكل عناصر SYMBOL التي تكون توابع العناصر التابعة الخاصة بعنصر PERIODIC_TABLE.

```
<xsl:template match="PERIODIC_TABLE/*/SYMBOL">
  <strong><xsl:value-of select="."/></strong>
</xsl:template>
```

في النهاية، كما وضح فيما سبق فإن علامة (/) نفسها تحدد نقط بدء الجذر للمستند ومثال لذلك تطبق هذه القاعدة على كافة عناصر PERIODIC_TABLE التي تكون عناصر الجذر للمستند.

```
<xsl:template match="/PERIODIC_TABLE">
  <html><xsl:apply-templates/></html>
</xsl:template>
```

تشير (/) إلى نقطة بدء الجذر وتشير (*/) إلى العنصر الجذري ومثال لذلك:

```
<xsl:template match="/*">
  <html>
    <head>
      <title>Atomic Number vs. Atomic Weight</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
```

مطابقة العناصر المنسدلة باستخدام علامة //

يكون في بعض الأحيان وبخاصة في الهياكل غير المتساوية من الأسهل تجاهل نقاط البدء المتوسطة وتحديد كافة عناصر نوع معطى سواء كانوا عناصر تابعة مباشرة أو توابع العناصر التابعة أو توابع العناصر التابعة وتشير علامة الخط المائل المزدوجة (//) إلى العناصر المنسدلة عند المستوى عشوائي ومثال لذلك تطبيق قاعدة القوالب هذه على كل العناصر المنسدلة NAME لعنصر PERIODIC_TABLE مهما كان عمقها.

```
<xsl:template match="PERIODIC_TABLE//NAME">
  <i><xsl:value-of select="."/></i>
</xsl:template>
```

يعتبر مثال الجدول الدوري بسيط للغاية ولكن هذه الخطوة تعتبر مهمة في الهياكل الطويلة خاصة عندما يحتوي عنصر على عناصر أخرى من نفس نوعه مثال لذلك يحتوي عنصر ATOM على ATOM.

يحدد عامل تشغيل علامة الخط المائل المزدوجة (//) في بداية النموذج أي عناصر منسدلة لنقطة بدء الجذر ومثال لذلك تقوم قاعدة القالب تلك معالجة ATOMIC_NUMBER كلها بينما تتجاهل مواقعها:

```
<xsl:template match="//ATOMIC_NUMBER">
  <i><xsl:value-of select="."/></i>
</xsl:template>
```

المطابقة باستخدام الرقم المعرف ID

قد يرغب المستخدم في تطبيق نمط محدد على عنصر واحد محدد دون تغيير كل العناصر الأخرى من نفس النوع وأبسط وسائل تنفيذ ذلك في XSL هو إلحاق نمط لسمة الرقم المعرف الخاصة بالعنصر وسيتم فعل ذلك باستخدام محدد ()id الذي يحتوي على قيمة الرقم المعرف في علامات اقتباس فردية ومثال ذلك فالقاعدة التالية تجعل العنصر الذي يحتوي على رقم معرف e47 سميكة.

```
<xsl:template match="id("e47")">
  <b><xsl:value-of select="."/></b>
</xsl:template>
```

يفترض هذا أن العناصر المراد تحديدها بهذا الأسلوب لها سمة معلن عنها كنوع ID في DTD الخاص بمستند المصدر وليست هذه هي القاعدة في معظم الحالات فمثلاً لا تملك العديد من المستندات DTD بل أنهم فقط مستندات صحيحة التكوين وليست صالحة وحتى إن كانت لهذه المستندات DTD فلا يوجد ضمان أن يكون لأي عنصر سمة نوع ID ويمكن استخدام عنصر xsl:key في ورقة النمط للإعلان عن أي سمات معينة في مستند الإدخال يجب معاملتهم على أنهم أرقام معرفة.

مطابقة السمات باستخدام @

تطابق علامة @ السمات وتحدد نقاط البدء وفقاً لاسم السمة وهذا وصحة الفصل الخامس وكل ما يجب فعله هو سبق السمة المراد تحديدها بعلامة @ ومثال لذلك توضح تعليمات البرمجة ١٤-٨ ورقة نمط والتي تقوم بإخراج جدول للإعداد الذرية في مقابل نقطة الانصهار ويكتب كلاً من قيمة عنصر MELTING_POINT وقيمة سمة UNITS الخاصة بها ويتم اختيارها بواسطة <xsl:value-of select="@UNITS"/>.

تعليمات البرمجة ١٤-٨: ورقة نمط XSL تحدد سمة UNITS باستخدام علامة @

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/PERIODIC_TABLE">
    <html>
```

```
    <body>
      <h1>Atomic Number vs. Melting Point</h1>
      <table>
        <th>Element</th>
        <th>Atomic Number</th>
        <th>Melting Point</th>
        <xsl:apply-templates/>
      </table>
    </body>
  </html>
</xsl:template>

<xsl:template match="ATOM">
   <tr>
    <td><xsl:value-of select="NAME"/></td>
    <td><xsl:value-of select="ATOMIC_NUMBER"/></td>
    <td><xsl:apply-templates select="MELTING_POINT"/></td>
  </tr>
</xsl:template>

<xsl:template match="MELTING_POINT">
  <xsl:value-of select="."/>
  <xsl:value-of select="@UNITS"/>
</xsl:template>

</xsl:stylesheet>
```

تذكر أن قيمة سمة نقطة البدء هي ببساطة قيمة السلسلة للسمة وبمجرد تطبيق ورقة النمط في تعليمات برمجة ١٤-٨ تظهر عناصر ATOM منسقة كما يلي:

```
<tr><td>Hydrogen</td><td>1</td><td>13.81Kelvin</td></tr>
<tr><td>Helium</td><td>2</td><td>0.95Kelvin</td></tr>
```

يمكن دمج السمات مع العناصر باستخدام عوامل تشغيل الهياكل المتعددة فمثلاً تشير BOILING_POINT/@UNITS إلى سمة UNITS لعنصر BOILING_ POINT وتطابق ATOM/*/@UNITS أي سمة UNITS لعنصر تابع من عناصر ATOM ويكون هذا ذو فائدة

عند المطابقة في مقابل سمات في قواعد القوالب ويجب تذكر أن ما تم مطابقته هو سـمة نقطـة البدء وليس العنصر الذي يحتويها ومن الأخطاء الشائعة التعارض الضمني بين نقطة بدء السـمة مع نقطة بدء العنصر التي تحتويها فمثلاً ما يلي هي قاعدة تحاول تطبيق القوالب على العنـاصر التابعة التي بها سمات UNITS:

```
<xsl:template match="ATOM">
  <xsl:apply-templates select="@UNITS"/>
</xsl:template>
```

ما يحدث بالفعل هو تطبيق القوالب على سمات UNITS الغير موجودة في عناصر ATOM.

تستخدم علامة (*) لتحديد كل سمات العنصر فمثـلاً BOILING_ POINT/@* تسـتخدم لتحديد كل سمات عناصر BOILING_POINT.

مطابقة التعليقات باستخدام comment ()

يجب تجاهل التعليقات في مستندات XML لأن جعل التعليقات جزء أساسي فـي المسـتند ليـس بالفكرة الجيدة ومع ذلك تقدم لغة XSL وسيلة لتحديد تعليق إذا تطلب الأمر ذلك.

لتحديد تعليق، استخدم النموذج comment() وعلى الرغم من وجود أقواس تشـبه أقـواس الدالة إلا أنه لا يتم وضع وسائط بداخلهم ولا يمكن التعبير بين التعليقات المختلفة ببساطة فمثـلاً يبدو عنصر DENSITY كما يلي:

```
<DENSITY UNITS="grams/cubic centimeter"><!- At 300K ->
  6.51
</DENSITY>
```

لا تقوم قاعدة الدالة بكتابة مخرجات قيمة الكثافة والوحدات فقـط ولكنـها تطبـع الظـروف المحيطة بقياس الكثافة وهي كما يلي:

```
<xsl:template match="DENSITY">
  <xsl:value-of select="."/>
  <xsl:value-of select="@UNITS"/>
  <xsl:apply-templates select="comment()"/>
</xsl:template>
```

السبب الوحيد لاستخدام تعليمات برمجة ١٤-١ تعليق لتحديد الظروف بدلاً من استخدام سـمة أو عنصر هو هذا المثال ولكن في الواقع العملي لا يجب أبداً وضع معلومات مهمة في تعليقـات والسبب الذي يجعل لغة XSL تتيح تحديد تعليقات هو جعل ورقة النمط تقوم بالتمويل مـن لغـة

ترميز إلى لغة ترميز أخرى بدون المساس بالتعليقات وأي استخدام مستند أصلي ضعيف التصميم والقاعدة التالية تطابق كل التعليقات وتقوم بنسخهم مرة أخرى باستخدام عنصر xsl:comment.

```
<xsl:template match="comment()">
  <xsl:comment><xsl:value-of select="."/></xsl:comment>
</xsl:template>
```

لاحظ أن القواعد الافتراضية المستخدمة لتطبيق القوالب لا تطبق علـى التعليقـات ولـذا إذا تطلب الأمر تنشيط القاعدة عندما يظهر التعليق يجب تضمين عنصـر xsl:apply-templates التي تحدد تعليقات في أي مكان تكون التعليقات موجودة فيه.

تستخدم عوامل التشغيل الهيكلية لتحديد تعليقات محددة ومثال ذلك تطـابق القـاعدة التاليـة التعليقات التي تقع داخل عناصر DENSITY

```
<xsl:template match="DENSITY/comment()">
   <xsl:comment><xsl:value-of select="."/></xsl:comment>
</xsl:template>
```

مطابقة تعليمات المعالجة باستخدام pi()

لا تعتبر التعليمات البرمجية أفضل من التعليقات عند كتابة مستندات XML الجيدة البنية المعتمـد عليها الممكن الحفاظ عليها ومع ذلك هناك استخدامات أخـرى تتضمـن إلحـاق أوراق نمـط المستندات

تحدد دالة pi() تعليمات المعالجة والوسائط الخاصة بدالة pi() هي سلسلة موضوعـة بيـن علامتي اقتباس تعطي اسم تعليمات المعالجة لتحديدها وإذا لك يتم تضمين وسائط يتم مطابقـة أول عنصر تابع للتعليمات المعالجة لنقطة البدء الخالية ومع ذلك يمكن اسـتخدام عوامـل التشـغيل الهيكلية ومثال لذلك تطابق هذه القاعدة أول عنصر تابع للتعليمات المعالجة لنقطة بـدء الجـذر وغالباً ما تكون تعليمات معالجة xml-stylesheet ويدرج عنصر xsl:pi element تعليمـات معالجة بها الاسم المحدود القيمة في مستند المخرجات.

```
<xsl:template match="/pi()">
  <xsl:pi name="xml-stylesheet">
   type="text/xsl" value="auto.xsl"
  </xsl:pi>
</xsl:template/>
```

تطابق هذه القاعدة تعليمات معالجة xml-stylesheet ولكن باسمها كما يلي:

```
<xsl:template match="pi("xml-stylesheet")">
  <xsl:pi name="xml-stylesheet">
```

```
    <xsl:value-of select="."/>
   </xsl:pi>
  </xsl:template/>
```

أحد الأسباب الرئيسية للتمييز بين عنصر الجذر ونقطة بدء الجذر حتى يمكن قراءة ومعالجة تعليمات المعالجة من التسجيل الأولي وعلى الرغم من استخدام تعليمات معالجة xml-stylesheet بناء جملة الاسم يساوي القيمة فإن لغة XSL لا تجعل بناء الجملة هذا سمات لأن تعليمات المعالجة ليست عناصر وتعتبر جهة قيمة تعليمات المعالجة هي أي قيمة ما بين المسافة البيضاء التي تتبع اسمها وعلامة الإعلام ?<.

لا تطابق القواعد الافتراضية المستخدمة لتطبيق القوالب تعليمات المعالجة لذا يجب تضمين عنصر xsl:apply-templates يطابق تعليمات المعالجة في الموقع المناسب في حالة تنشيط تلك القاعدة عند مواجهة تعليمات معالجة.xml-stylesheet ومثال لذلك تطبق قاعدة القالب هذه لنقطة بدء الجذر قوالب لتعليمات المعالجة كما يلي:

```
  <xsl:template match="/">
    <xsl:apply-templates select="pi()"/>
    <xsl:apply-templates select="*"/>
  </xsl:template>
```

مطابقة نقاط بدء النص باستخدام text()

يتم تضمين قيم نقاط بدء النص كجزء من قيمة العنصر المحدد على الرغم من اعتبارها نقاط بدء ومع ذلك يتيح عامل تشغيل text() تحديد عنصر النص التابع للعنصر الأساسي بدقة وبالرغم من وجود الأقواس فلا يقبل عامل التشغيل وسائط ومثال لذلك:

```
  <xsl:template match="SYMBOL">
    <xsl:value-of select="text()"/>
  </xsl:template>
```

السبب الرئيسي في وجود عامل التشغيل هو القواعد الافتراضية وتقدم معالجات XSL القاعدة الافتراضية التالية سواء حددها الناشر أم لا:

```
  <xsl:template match="text()">
    <xsl:value-of select="."/>
  </xsl:template>
```

يعني هذا أنه في حالة تطبيق قالب على نقطة بدء النص يصبح نص نقطة البدء من المخرجات ويمكن التجاوز عن القواعد الافتراضية إذا رغب المستخدم في ذلك فمثلاً لمنع وتضمين قاعدة القالب الفارغ التالية في ورقة النمط نقاط بدء النص من أن تصبح مخرجات إلا في حالة مطابقتها بصورة محددة بواسطة قاعدة أخرى.

```
<xsl:template match="text()">
</xsl:template>
```

استخدام عامل التشغيل "أو" ويرمز له بعلامة |

يتيح الشريط العمودي | لقاعدة القالب أن تطابق نماذج متعددة وفي حالة تطابق نقطة مع نمــوذج أو أخر فإنه سينشط القالب وما يلي قاعدة قالب تطابق كلاً من عنـــاصر ATOMIC_NUMBER وATOMIC_WEIGHT.

```
<xsl:template match="ATOMIC_NUMBER|ATOMIC_WEIGHT">
  <B><xsl:apply-templates/></B>
</xsl:template>
```

يمكن تضمين مسافات بيضاء حول علامة | إذا كان ذلك سيجعل التعليمات البرمجية أوضـــح وما يلي يوضح ذلك.

```
<xsl:template match="ATOMIC_NUMBER | ATOMIC_WEIGHT">
  <B><xsl:apply-templates/></B>
</xsl:template>
```

يمكن استخدام أكثر من نموذجين على التوالي فمثلاً قاعدة القالب التالية تطبق عناصر علـــى ATOMIC_NUMBER وATOMIC_WEIGHT وSYMBOL وهذا يعني أنها تطـــابق عنـــاصر ATOMIC_NUMBER وATOMIC_WEIGHT وSYMBOL:

```
<xsl:template match="ATOMIC_NUMBER | ATOMIC_WEIGHT |
SYMBOL">
  <B><xsl:apply-templates/></B>
</xsl:template>
```

يتم تقييم عامل تشغيل علامة / قبل علامة تشغيل | ولهذا تحدد قاعدة القالب التالية عنصـــر تابع ATOMIC_NUMBER لعنصر ATOM أو ATOMIC_WEIGHT غير محدد الأصل وليس تابع ATOMIC_NUMBER لعنصر ATOM أو تابع ATOMIC_WEIGHT لعنصر ATOM.

```
<xsl:template match="ATOM/ATOMIC_NUMBER|ATOMIC_WEIGHT">
  <B><xsl:apply-templates/></B>
</xsl:template>
```

أداء اختبارات باستخدام علامة [ ]

كافة الاختبارات التي نفذت إلى الآن هي لمعرفة ظهور نقاط بدء مختلفة من عدمه وعامة يمكـــن إجراء المزيد من الاختبارات لمعرفة مزيد من التفاصيل بشأن نقاط البدء التي تطـــابق نمـــوذج باستخدام علامة [ ] ومن الاختبارات الممكن تنفيذها ما يلي:

- إذا كان عنصر يحتوي على عنصر تابع أو سمة أو نقطة بدء أخرى.
- إذا كان قيمة سمة هي سلسلة معطاة.
- إذا كان قيمة عنصر تطابق سلسلة.
- أي موقع تقع نقطة بدء معطاة في الهيكل.

يتم تكوين عنصر ١٠٦، seaborgium وكميات ضئيلة جداً وأطول مدة يحياها نظيره هي مدة ٢٠ ثانية ولذا مع وجود عنصر يصعب تكوينه ومدة حياته طويلة فمن الصعب قياس الكثافة أو نقطة الانصهار أو أي خصائص حجم أحرف وبالتالي يلغي مستند الجدول الدوري العناصر التي تصف خصائص الحجم لعنصر seaborgium والذرات المماثلة كنتيجة لعدم وجود بيانات كاملة فإذا أردت أن تنشئ جدول فيه الأرقام الذرية مقابل نقطة الانصهار. يجب إلغاء العناصر التي ليس لها نقطة انصهار معروفة ولفعل ذلك يمكن تحديد مطابقة في مقابل عناصر ATOM التي لها عناصر تابعة هي MELTING_POINT مثلما يوضح ما يلي:

```
<xsl:template match="ATOM[MELTING_POINT]">
  <tr>
    <td><xsl:value-of select="NAME"/></td>
    <td><xsl:value-of select="MELTING_POINT"/></td>
  </tr>
</xsl:template>
```

لاحظ أن ملقم مطابقته هو عنصر ATOM وليس عنصر MELTING_POINT كما في حالة ATOM/MELTING_POINT.

تحتوي أقواس الاختبار على أكثر من اسم عنصر تابع فقد تحتوي على أي تعبير محدد وتعتبر التغييرات المحددة مجموعة أكبر لنماذج المطابقة التي سيتم مناقشتها في المقطع التالي وإذا كان للعنصر المحدد عنصر تابع يطابق هذا التعبير فهو بذلك يطابق النموذج النهائي ومثال على ذلك تطابق قاعدة القالب عناصر ATOM التي لها تابع NAME أو تابع SYMBOL.

```
<xsl:template match="ATOM[NAME | SYMBOL]">
</xsl:template>
```

تستخدم قاعدة القالب علامة (*) لتطابق أي عنصر يحتوي على تابع NAME.

```
<xsl:template match="*[NAME]">
</xsl:template>
```

تطابق قاعدة القالب التالية عناصر ATOM التي لها تابع DENSITY الذي يكون له سمة UNITS.

```
<xsl:template match="ATOM[DENSITY/@UNITS]">
</xsl:template>
```

يمكن الرجوع إلى مثال سابق لإيجاد كل العناصر التابعة التي لها سـمات UNITS بطريقة مفصلة استخدام علامة (*) لإيجاد كل العناصر و[@UNITS] للوصول إلى العناصر التي لـها سمات UNITS ومثل ما يلي:

```
<xsl:template match="ATOM">
  <xsl:apply-templates select="*[@UNITS]"/>
</xsl:template>
```

إحدى أنواع اختيار النماذج التي أثبتت فائدتها هي معادلة السلسلة وتختبر علامة المساواة (=) ما إذا كانت قيمة نقطة البدء تطابق بصورة مشابهة سلسلة معطاة ومثال لذلك يجد القالب التـالي عنصر ATOM الذي يحتوي على عنصر ATOMIC_NUMBER التي تتضمن محتويات سلسـلة ١٠ وهي غاز النيون.

```
<xsl:template match="ATOM[ATOMIC_NUMBER='10']">
  This is Neon!
</xsl:template>
```

تبدو الاختبارات في مقابل محتوى العنصر ماهرة للغاية لأنه يجب إيجاد القيمة صحيحة تماماً بما في ذلك المسافات البيضاء وقد يجد المستخدم الاختبار في مقابل قيم السمات أسهل حيث أنـه ليس من المحتمل أن يوجد لهم مسافات بيضاء غير ضرورية وتوضح ورقة النمط في تعليمـات برمجة ١٤-٩ تطبق القوالب فقط على عناصر ATOM التي تكون قيمة سمة STATE الخاصـة بهم هي الثلاثة حروف GAS.

تعليمات البرمجة ١٤-٩: ورقة نمط XSL تحدد عناصر ATOM التي تكون قيمة سمة STATE الخاصة بهم هي GAS

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

 <xsl:template match="PERIODIC_TABLE">
  <html>
   <head><title>Gases</title></head>
   <body>
    <xsl:apply-templates select="ATOM[@STATESTATE='GAS']"/>
   </body>
  </html>
```

```
</xsl:template>

<xsl:template match="ATOM">
  <P><xsl:value-of select="."/></P>
</xsl:template>

</xsl:stylesheet>
```

يمكن استخدام تعبيرات XSL أخرى ويناقشها الجزء التالي لمزيد من المطابقات المعقدة ومثال لذلك يمكن تحديد كل العناصر التي تبدأ أسمائهم بحرف أ "A" أو كل العناصر التي يكون الرقم الذري لها أقل من ١٠٠.

تعبيرات لتحديد نقاط البدء

تستخدم سمة select في كل من xsl:apply-templates وxsl:value-of وxsl:for-each وxsl:copy-of وxsl:sort لتحديد أي نقاط البدء التي يتم العمل فيها وقيمة هذه السمة هي تعبير والتغييرات هي مجموعة إضافية على نماذج المطابقة التي تمت مناقشتها في الجزء السابق وهذا يعني أن كل نماذج المطابقة هي تغييرات تحديد ولكن ليس كل تعبيرات التحديد هي نماذج مطابقة وتتيح نماذج المطابقة مطابقة نقاط البدء باستخدام اسم العنصر والعناصر التابعة والمشتقات والسمات وأيضاً بإجراء اختبارات بسيطة على تلك العناصر بينما تتيح تعبيرات التحديد تحديد نقاط بدء عن طريق كل هذه المعايير وأيضاً عن طريق الإشارة إلى عناصر الأصل وعناصر siblings وإجراء المزيد من الاختبارات المعقدة وبالإضافة إلى ذلك فالاختبارات ليست مقيدة فقط في إنتاج قائمة بنقاط البدء ولكن يمكن أيضاً إنتاج أرقام وسلاسل وعمليات منطقية.

محاور نقطة البدء

لا يتم تقييد التعبيرات في تحديد العناصر التابعة والمشتقات لنقطة البدء الحالية حيث تقدم لغة XSL عدد من المحاور تستخدم للتحديد من أجزاء مختلفة من الشجرة على أن تكون تلك التحديدات مرتبطة بنقطة البدء الحالية وغالباً ما تكون نقطة البدء التي يطابقها القالب ويلخص جدول ١٤-٢ تلك المحاور ومعاينها.

الجدول ١٤-٢

محاور التعبير

المحور	يحدد من
from-ancestors()	يحدد من أصل نقطة البدء الحالية صعوداً إلى نقطة بدء الجذر
From-ancestors or-self()	يحدد من نقطة البدء الحالية نفسها وأصول نقطة البدء الحالية
from-attributes()	يحدد من سمات نقطة البدء الحالية
from-children()	يحدد من التوابع المباشرة لنقطة البدء الحالية
from-descendants()	يحدد من التوابع لنقطة البدء الحالية وتوابع توابع نقطة البدء الحالية .....الخ
from-descendants-or-self()	يحدد من نقطة البدء الحالية ومشتقاتها
from-following()	يحدد كل نقاط البدء التي تبدأ بعد نهاية نقطة البدء الحالية
from-following-siblings()	يحدد كل نقاط البدء التي تبدأ بعد نهاية نقطة البدء الحالية ويكون لها نفس أصل نقطة البدء الحالية
from-parent()	يحدد نقطة بدء الأصل الوحيدة لنقطة البدء الحالية
from-preceding()	يحدد كل نقاط البدء التي تبدأ قبل بداية نقطة البدء الحالية
from-preceding siblings()	يحدد كل نقاط البدء التي تبدأ قبل بداية نقطة البدء الحالية ويكون لها نفس أصل نقطة البدء الحالية
from-self()	يحدد نقطة البدء الحالية



محاور from-following وfrom-preceding موضع تساؤلات كثيرة وقد لا يتم تضمينها في النسخة النهائية من لغة XSL وإذا تم تضمينها قد يتغير معناها.

تؤدي تلك المحاور مهمة الدوال التي تحدد من مجموعه نقاط البدء الموضحة في العمود الثاني في جدول ١٤-٢ وتحتوي الأقواس على تعبيرات تحديد للوصول إلى أسفل قائمة نقاط البدء وكما توضح قاعدة القالب التالية فقد تحتوي على اسم العنصر المراد تحديده.

```
<xsl:template match="ATOM">
 <tr>
  <td>
   <xsl:value-of select="from-children(NAME)"/>
  </td>
  <td>
   <xsl:value-of select="from-children(ATOMIC_NUMBER)"/>
  </td>
  <td>
   <xsl:value-of select="from-children(ATOMIC_WEIGHT)"/>
  </td>
 </tr>
</xsl:template>
```

تطابق قاعدة القالب عناصر ATOM وعند مطابقة عنصر ATOM يتم تحديد عنصـر NAME وعنصر ATOMIC_NUMBER وكذلك عنصر ATOMIC_WEIGHT من العنـاصر التابعـة لعنصر ATOM الذي تم مطابقته وتكون المخرجات في شكل خلايا جدول وإذا كان هناك أحد تلك العناصر المرغوبة له أكثر من عنصر تابع واحد مثل أن يكون للعنصر ثلاثة عناصر NAME يتم تحديد أول عنصر فقط.

لا يتيح محور ()from-children تنفيذ أي شيء لا تستطيع تنفيذه بواسطة أسماء العنـاصر وحدها وفي الواقع تعتبر "select="ATOMIC_WEIGHT ولكن المحاور الأخرى لـها تأثـير مثير وفعال.

لا تسمح نماذج المطابقة بالإشارة إلى عناصر الأصل على عكس تعبيرات التحديـد يسـتخدم محور ()from-parent للإشارة إلى الأصل فمثلاً تقوم قاعدة التالية بوضع مخرجات قيم الذرات التي لها عنصر BOILING_POINT تابع.

```
<xsl:template match="ATOM/BOILING_POINT">
 <P><xsl:value-of select="from-parent(ATOM)"/></P>
</xsl:template>
```

في المثال تم مطابقة عنصر BOILING_POINT التابع ولكن عنصر الأصل ATOM كـان من المخرجات.

بعض الذرات المشعة مثل البولونيوم لها أنصاف حياة قصيرة جداً حتى أن خصائص الحجـم مثل نقطـة الغليـان ونقطـة الانصـهار لا يمكـن قياسـها وبالتـالي ليـس لكـل عنـاصر BOILING_POINT تابعة وتتيح القاعدة السابقة وضع العناصر التي لها نقطة غليان فعلياً فـي

المخرجات وبالتوسع في هذا المثال تطابق تعليمات برمجة ١٤-١٠ عناصر MELTING_POINT ولكن فعلياً يتم إخراج عنصر ATOM الأصل باستخدام from-parent(ATOM).

تعليمات البرمجة ١٤-١٠: ورقة نمط تضع كمخرجات العناصر التي لها نقاط انصهار معروفة

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/">
   <html>
    <body>
     <xsl:apply-templates select="PERIODIC_TABLE"/>
    </body>
   </html>
  </xsl:template>

  <xsl:template match="PERIODIC_TABLE">
   <h1>Elements with known Melting Points</h1>
Continued
Listing 14-10ت(continued)
     <xsl:apply-templates select="//MELTING_POINT"/>
  </xsl:template>

  <xsl:template match="MELTING_POINT">
   <p>
    <xsl:value-of select="from-parent(ATOM)"/>
   </p>
  </xsl:template>

</xsl:stylesheet>
```

في بعض الأحيان، قد يرغب المستخدم في تحديد أقرب أصل لعنصر لنوع معطى وتؤدي دالة from-ancestors() ذلك فمثلاً تدرج تلك القاعدة قيمة أقرب عنصر PERIODIC_TABLE الذي يحتوي على عنصر SYMBOL الذي تم مطابقته.

```
<xsl:template match="SYMBOL">
  <xsl:value-of select="from-ancestors(PERIODIC_TABLE)"/>
</xsl:template>
```

منهج دالة ()from-ancestors-or-self هو نفسه منهج دالة ()from-ancestors إلا في حالة تطابق نقطة البدء الحالية مع نوع الوسيطة ففي تلك الحالة يتم إرجاعها "نقطة البدء الحالية" بدلاً من الأصل الحقيقي فمثلاً القاعدة التالية تطابق كل العناصر فإذا كانت العناصر المطابقة هي PERIODIC_TABLE فإن هذا PERIODIC_TABLE نفسه يتم تحديده في xsl:value-of.

```
<xsl:template match="*">
<xsl:value-of select="from-ancestors-or-self(PERIODIC_TABLE)"/>
</xsl:template>
```

بالإضافة إلى اسم نقطة البدء وأحرف البدل قد تكون وسائط دالة ()from-axis إحدى أربع دوال نوع نقطة البدء التالية:

- comment()
- text()
- pi()
- node()

يحدد نوع نقطة بدء ()comment نقطة بدء التعليق ويحدد نوع نقطة بدء ()text نقطة بدء النص ويحدد نوع نقطة بدء ()pi نقطة بدء تعليمات المعالجة وتحدد نوع نقطة بدء ()node أي نوع من أنواع نقاط البدء وتحدد علامة أحرف البدل عناصر نقاط البدء وتحتوي نوع نقطة البدء ()pi على وسائط اختيارية تحدد الاسم لتعليمات المعالجة المراد تحديدها.

تشتمل القاعدة التالية على قيمة عنصر ATOM المتطابقة في عنصر P باستخدام -from ()self مع نوع نقطة بدء ()node.

```
<xsl:template match="ATOM">
  <P><xsl:value-of select="from-self(node())"/></P>
</xsl:template>
```

في هذا المثال لا يتشابه تحديد (()from-self(node مع تحديد ATOM وتحاول القاعدة التالية الحصول على قيمة تابع ATOM لعنصر ATOM وتلك ليست قيمة عنصر ATOM المطابق ولكن قيمة عنصر ATOM مختلف وهو تابع لعنصر ATOM المطابق.

```
<xsl:template match="ATOM">
  <P><xsl:value-of select="ATOM"/></P>
</xsl:template>
```

عوامل التشغيل الهيكلية

تستخدم عوامل تشغيل الخط المائل / والخط المائل المزدوج // لوصل تعبيرات محددة معاً وتقوم تعليمات برمجة ١٤-١١ بطباعة جدول للأسماء العناصر والأعداد الذرية ونقاط الانصهار للعناصر التي لها نقاط انصهار وهي تؤدي ذلك بتحديد أصل عنصر MELTING_POINT ثم إيجاد عناصر NAME وATOMIC_NUMBER التابعة للأصل باستخدام select="from-parent(*)/from-children(NAME)".

تعليمات البرمجة ١٤-١١: جدول بنقاط الانصهار يقابله العدد الذري

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

   <xsl:template match="/PERIODIC_TABLE">
    <html>
      <body>
        <h1>Atomic Number vs. Melting Point</h1>
        <table>
          <th>Element</th>
          <th>Atomic Number</th>
          <th>Melting Point</th>
          <xsl:apply-templates select="from-children(ATOM)"/>
        </table>
      </body>
    </html>
   </xsl:template>

   <xsl:template match="ATOM">
<xsl:apply-templates
      select="from-children(MELTING_POINT)"/>
   </xsl:template>
```

```
<xsl:template match="MELTING_POINT">
  <tr>
   <td>
     <xsl:value-of
      select="from-parent(*)/from-children(NAME)"/>
   </td>
   <td>
     <xsl:value-of
   select="from-parent(*)/from-children(ATOMIC_NUMBER)"/>
   </td>
   <td>
     <xsl:value-of select="from-self(*)"/>
     <xsl:value-of select="from-attributes(UNITS)"/>
   </td>
  </tr>
</xsl:template>

</xsl:stylesheet>
```

هناك طرق أخرى لحل تلك المشكلة فيمكن استخدام محـــور from-preceding-siblings() ومحور from-following-siblings() أو كلاهما معاً في حالة أن يكون الموقع المتصـــل بـــهم سواء السابق أو اللاحق غير مؤكد وستبدو قاعدة القالب لعنصر MELTING_POINT كما يلي:

```
<xsl:template match="MELTING_POINT">
  <tr>
   <td>
    <xsl:value-of
    select="from-preceding-siblings(NAME)
         | from-following-siblings(NAME)"/>
   </td>
   <td>
    <xsl:value-of
    select="from-preceding-siblings(ATOMIC_NUMBER)
         | from-following-siblings(ATOMIC_NUMBER)"/>
   </td>
```

```
    <td>
      <xsl:value-of select="from-self(*)"/>
      <xsl:value-of select="from-attributes(UNITS)"/>
    </td>
  </tr>
</xsl:template>
```

بناء الجملة المختصر

تعتبر دوال ()from-axis المتعددة في جدول ١٤-٢ كثيرة الكلمات مما يجعل كتابتها معرض الأخطاء ولذلك تقدم لغة XSL بناء جملة مختصر يمكن استبداله بمعظم المحاور الشائعة ويتم استخدام بناء الجملة المختصر هذا بصورة أكبر في الواقع العملي ويوضح جدول ١٤-٣ بناء الجملة الكامل وبناء الجملة المختصر الذي يوازيه.

الجدول ١٤-٣

بناء الجملة المختصر للتعبيرات المحددة

بناء الجملة الكامل	الاختصار
from-self(node())	
from-parent(node())	..
from-children(name)	Name
from-attributes(name)	@name
/from-descendants-or-self(node())	//

توضح تعليمات برمجة ١٤-١٢ بناء الجملة المختصر بواسطة إعادة كتابة تعليمات برمجة ١٤-١١ باستخدام بناء الجملة المختصر والمخرجات الناتجة في كلتا الحالتين أي باستخدام ورقي النمط هما نفسهما بدون تغيير.

تعليمات البرمجة ١٤-١٢ : جدول لنقطة الانصهار في مقابل العدد الذري باستخدام بناء الجملة المختصر

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
```

```
<xsl:template match="/PERIODIC_TABLE">
  <html>
    <body>
      <h1>Atomic Number vs. Melting Point</h1>
      <table>
        <th>Element</th>
        <th>Atomic Number</th>
        <th>Melting Point</th>
        <xsl:apply-templates select="ATOM"/>
      </table>
    </body>
  </html>
</xsl:template>

<xsl:template match="ATOM">
    <xsl:apply-templates
    select="MELTING_POINT"/>
  </xsl:template>

  <xsl:template match="MELTING_POINT">
    <tr>
     <td>
       <xsl:value-of
        select="../NAME"/>
     </td>
     <td>
       <xsl:value-of
      select="../ATOMIC_NUMBER"/>
     </td>
     <td>
       <xsl:value-of select="."/>
       <xsl:value-of select="@UNITS"/>
     </td>
```

```
    </tr>
  </xsl:template>

</xsl:stylesheet>
```

تستخدم نماذج المطابقة فقط بناء الجملة المختصر وليس ذلك بينما بناء الجملة الكامل باستخدام دوال from-axis() الموجودة في جدول ١٤-٢ محدودة في تعبيرات التحديد.

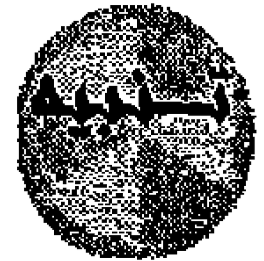
أنواع التعبيرات

يتم تقييم كل تعبير إلى قيمة واحدة فمثلاً التعبير ٣+٢ يتم تقييمه إلى القيمة ٥ ويوجد خمس أنواع من التعبيرات المستخدمة فيما سبق يتم تقييمها إلى مجموعات نقاط بدء ويوجد ضمن أنواع التعبيرات في لغة XSL وهم:

- مجموعات نقاط البدء
- العمليات المنطقية
- الأرقام
- السلاسل
- أجزاء شجرة النتائج

مجموعات نقاط البدء

تعتبر مجموعة نقاط البدء قائمة بنقاط البدء من مستند الإدخال وترجع دوال from-axis() الموجودة في جدول ١٤-٢ مجموعة نقاط البدء التي تحتوي على نقطة البدء المتطابقة معها وتعتمد أي نقاط البدء تكون في مجموعة نقاط البدء التي ترجعها إحدى هذه الدوال على نقطة البدء الحالية والمعروفة أيضاً كنقط بدء السياق وعلى وسيطة الدالة وكذلك على أي الدوال هي.



يعتبر المبرمجين الذين اعتادوا على استخدام اللغات الموجهة للكائن مثل Java و++C نقطة البدء الداخلية على أنها الكائن الذي يضع الدالة موضع التنفيذ فمثلاً في a.doSomething(b, c) تعتبر نقط البدء الحالية هي a ومع ذلك ففي لغة XSL تكون نقطة البدء دائماً ضمنية وتكتب مثل doSomething(b, c) كما قد يحدث في ملف تعريف فئة a.

مثال على ذل، يرجع تعبير "select= from-children(ATOM)" مجموعــــة نقـــاط بـــدء تحتوي على كل من عناصر ATOM في هذا المستند عندما تكون نقطة البدء الحالية هي عنصــر PERIODIC_TABLE فـــــي مثـــــال ١٤-١ ويرجــــــع تعبــــــير "select="from-children(ATOM)/from-children(NAME)" مجموعة نقاط البدء التي تحتوي على عنصــر نقطي البدء <NAME>Hydrogen</NAME> و <NAME>Helium</NAME> وذلك عندما تكون نقط بدء السياق هي عنصر PERIODIC_TABLE في مثال ١٤-١.

تعتبر نقطة بدء السياق عضو في قائمة نقاط بدء السياق وقائمة نقاط بدء السياق هي مجموعة العناصر التي تطابق نفس القاعدة في نفس الوقت وغالباً يكون هذا التطابق نتيجة وجـــود أحــد العنصرين إما xsl:apply-templates أو xsl:for-each فمثلاً عند تطبيق تعليمـــات برمجــة ١٤-١٢ على تعليمات برمجة ١٤-١ فإن قالب ATOM يوضع موضع التنفيذ مرتين، مرة لـــذرة الهيدروجين ومرة لذرة الهليوم أول مرة يتم وضعها موضع التنفيذ فيها تكون نقطة بدء الســـياق هي عنصر ATOM الهليوم وفي كلتا الخالتين فإن قائمة نقط بدء السياق هي المجموعــــة التـــي تحتوي على كل من عناصر ATOM للهليوم والهيدروجين.

يدرج جدول البدء ١٤-٤ عدد من الدوال التي تعمل على مجموعات نقاط البدء سواء كوسائط أو كنقطة بدء السياق.

الجدول ١٤-٤

الدوال التي تعمل على مجموعات نقاط البدء

الدالة	نوع الاسترجاع	ترجع
Function	أرقام	موقع نقطة بدء السياق في قائمة نقط بدء السياق وأول نقطة بدء في القائمة يكون مركزها.
position()	أرقام	رقم نقاط البدء في مجموعة نقطة بدء السياق.
Count(node-set) number	أرقام	رقم نقاط البدء في node set
id(string)	مجموعـــة نقـــاط البدء	مجموعة نقاط البدء التي تحتوي على عنصـــر واحد في أي مكان في نفس المستند الذي يكـون رقم المعرف (ID) هـــو string أو مجموعـــة فارغة إذا لم يكن لعنصر رقم معرف ID محدد.

الجدول ٤-١٤

الدوال التي تعمل على مجموعات نقاط البدء

الدالة	نوع الاسترجاع	ترجع
Idref(node-set)	مجموعة نقاط البدء	مجموعة نقاط البدء التي تحتوي على كل العناصر في المستند التي تكون الرقم المعرف لها إحدى العلامات التي يفصلها مسافات بيضاء في قيم نقاط البدء في وسيطه node set.
Key(string name, string value)	مجموعة نقاط البدء	مجموعة نقطة البدء تحتوي على كل نقاط البدء في هذا المستند على أن يكون لها مفتاح بقيمة محددة ويتم إعداد تلك المفاتيح بعنصر xsl:key الأعلى مستوى.
Keyref(string name, node set values)	مجموعة نقاط البدء	مجموعة نقطة البدء تحتوي على كل نقاط البدء في هذا المستند الذي له مفتاح تكون قيمته نفس قيمة أحد نقاط البدء في الوسطة الثانية.
doc(string URI)	مجموعة نقاط البدء	مجموعة نقطة البدء في المستند أو جزء يشار إليه بواسطة URI ويتم اختيار نقاط البدء من ارتساء الأسماء أو Xpointer المستخدم بواسطة URI ويكون عنصر الجذر اسم المستند في مجموعة نقاط البدء وURIs النسبية ترتبط بنقطة البدء الحالية الموجودة في مستند الإدخال.
docref(node set)	مجموعة نقاط البدء	مجموعة نقطة بدء تحتوي على كل نقاط البدء في مستند أو أكثر وتشير URI إليه على أنه قيمة لوسيطه node set
local-part (node set)	سلسلة	يستخدم الجزء المحلي وهو الجزء بادئة حيز الاسم لنقطة البدء الأولى في وسيطه node set دون وسائط للحصول على الجزء المحلي لنقطة بدء السياق.

الجدول ١٤-٤

الدوال التي تعمل على مجموعات نقاط البدء

الدالة	نوع الاسترجاع	ترجع
Namespace(node set)	سلسلة	تستخدم URI الخاصة بحيز الاسم لنقطة البدء الأولى في مجموعة نقطة البدء دون أي وسائط للحصول على URI الخاص بحيز الاسم لنقطة بدء السياق وترجع سلسلة فارغة إذا كانت نقطة البدء في حيز الاسم الافتراضي.
Qname(node set)	سلسلة	يستخدم الاسم المؤهل وهو كلاً من البادئة والجزء المحلي لنقطة البدء الأولى في وسيطه node set دون أي وسائط للحصول على الاسم المؤهل لنقطة بدء السياق.
Generate-id (node set)	سلسلة	يستخدم معرف فريد لنقطة البدء الأولى في وسيطه node set دون وسائط لإيجاد رقم معرف ID لنقطة بدء السياق.


يناقش الفصل ١٨ "Namespace" حيز أسماء URIs، البادئة أو الأجزاء المحلية.


تعتبر دوال ()doc و()docref غامضة بعض الشيء خاصة إذا كانت URIs تشير إلى أجزاء من نقطة البدء أو بيانات لا تعتبر مستند XML صحيح التكوين وتبقى التفاصيل في حاجة "إلى ترتيب في الإصدارات المستقبلية لمواصفات XSL.

إذا وضعت وسيطة من نوع خطأ لإحدى هذه الدوال فستحاول لغة XSL تحويل الوسيطة إلى نوع صحيح مثلاً عن طريق تحويل رقم ١٢ إلى سلسلة "١٢" ومع ذلك فلا يتم تحويل وسائط إلى مجموعات نقاط بدء.

تستخدم دالة ()position لحساب العناصر وتعليمات برمجة ١٤-١٣ هي ورقة نمط تسبق اسم كل ذرة بموقعها في المستند باستخدام:

using <xsl:value-of select="position()"/>.

تعليمات البرمجة ١٤-١٣: ورقة نمط ترقم الذرات بالترتيب الذي يظهروا به في المستند

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/PERIODIC_TABLE">
    <HTML>
      <HEAD><TITLE>The Elements</TITLE></HEAD>
      <BODY>
        <xsl:apply-templates select="ATOM"/>
      </BODY>
    </HTML>
  </xsl:template>

  <xsl:template match="ATOM">
    <P>
      <xsl:value-of select="position()"/>.
      <xsl:value-of select="NAME"/>
    </P>
  </xsl:template>

</xsl:stylesheet>
```

عند تطبيق ورقة النمط السابقة على تعليمات برمجة ١٤-١ تكون المخرجات كما يلي:

```
<HTML><HEAD><TITLE>The
Elements</TITLE></HEAD><BODY><P>1.
   Hydrogen</P><P>2.
   Helium</P></BODY></HTML>
```

العمليات المنطقية

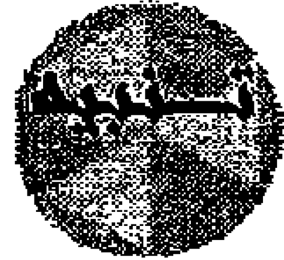
تحتمل العمليات المنطقية إحدى قيمتين إما صحيحة أو خطأ وتتيح بطريقة ضمنيـة عندمـا يتـم استخدام سلسلة أو رقم أو مجموعة نقطة بدء في مكان استخدام العمليات المنطقية كما في سـمة

test لعنصر xsl:if يمكن أداء تلك التحولات بواسطة دالة boolean() والتي تحول وسيطة لعمليات منطقية طبقاً للقواعد التالية:

- ويعتبر رقم خطأ إذا كانت قيمته صفر أو NaN وهو رمز يعني أن القيمة ليست رقم Not a Number وتستخدم لنتيجة القسمة على صفر والعمليات غير المسموح بها وتكون قيمة الرقم في أي حالة أخرى صحيحة.
- تعتبر قيمة مجموعة نقطة بدء فارغة خطأ وكل قيم مجموعات نقاط البدء الأخرى صحيحة.
- تعتبر قيمة سلسلة طولها صفر خطأ وقيمة كافة السلاسل الأخرى صحيحة.

تتيح العمليات المنطقية كنتيجة عن التغييرات المتضمنة في العمليات التالية:

= المساواة
< أصغر من "حقيقة" <
> أكبر من
<= أصغر من أو يساوي "حقيقة" <=
>= Hكبر من أو يساوي



لا يسمح بوجود علامة > أصغر من في قيم السمات وبالتالي يجب استبدالها بعلامة < حتى في حالة استخدامها كعامل تشغيل أصغر من.

تستخدم عوامل التشغيل السابقة بطريقة شائعة في الاختبارات التمهيدية لتحديد ما إذا كانت قاعدة قد وضعت موضع تنفيذ أم لا وتحتوي تعبيرات التحديد ليس فقط على نماذج تحدد نقاط بدء معينة ولكن أيضاً على اختبارات تمهيدية أخرى تقوم بتصفية قائمة نقاط البدء المحددة فمثلاً تحدد from-children(ATOM) كل العناصر التابعة لعنصر ATOM في نقطة البدء الحالية وتحدد from-children(ATOM[position()=1]) العنصر التابع الأول فقطط لعنصر ATOM في نقطة البدء الحالية وتعتبر [position()=1] اختيار تمهيدي لاختيار نقطة بدء ATOM التي تقوم بإرجاع نتيجة العمليات المنطقية وتكون صحيحة إذا كانت موقع نقطة البدء الحالية يساوي واحد وتكون خطأ في أي حالة أخرى ويكون لأي اختيار نقطة بدء أي من الاختيارات التمهيدية "المعرفات" وعموماً فإن وجود أكثر من واحد يعتبر شيء غير عادي.

المثال التالي يوضح تطبيق قاعدة القالب على عنصر ATOM الأول في الجدول الدوري وليس على عناصر تالية عن طريق اختبار ما إذا كان موقع العنصر يساوي ١.

```
<xsl:template match="PERIODIC_TABLE/ATOM[position()=1]">
  <xsl:value-of select="."/>
</xsl:template>
```

تطبق قاعدة القالب على كل عناصر ATOM التي لا تكون العنصر التابع الأول لعنصر PERIODIC_TABLE عن طريق اختبار ما إذا كان الموقع أكبر من واحد.

```
<xsl:template match="PERIODIC_TABLE/ATOM[position()>1]">
  <xsl:value-of select="."/>
</xsl:template>
```

تدمج الكلمات الأساسية and وor بطريقة منطقية تعبيرين من تعبيرات العمليات المنطقية وفقاً للقواعد العادية للعمليات المنطقية فمثلاً يمكن افتراض أن المطلوب هو تطبيق قالب على عنصر ATOMIC_NUMBER الذي يكون أول وأخر تابع للعنصر الأصل وهذا يعني أنه العنصر الوحيد لأصله ويستخدم هذا القالب and وتكون النتيجة هي ما يلي:

```
<xsl:template
 match="ATOMIC_NUMBER[position()=1 and position()=last()]">
  <xsl:value-of select="."/>
</xsl:template>
```

تطبق قاعدة القالب على كلاً من أول وأخر عناصر ATOM في أصلهم عن طريق المطابقة عندما يكون الموقع واحد أو يكون الموقع الأخير.

```
<xsl:template match="ATOM[position()=1 or position()=last()]">
  <xsl:value-of select="."/>
</xsl:template>
```

استخدم المثال السابق فهو عملية منطقية "أو" ولذا فإنها تطابق في حالة أن يكون كلا الشرطين صحيحين وهذا يعني أنها ستطابق ATOM التي تكون أول وأخر تابع للأصل.

لا تكن توجد كلمة أساسية تسمى not في لغة XSL ولكن هناك حالة not() وتقوم هذه الدالة بعكس العملية عن طريق تضمينها في دالة not() ويوضح المثال التالي تحديد القالب لكل عناصر ATOM التي لا تكون أول تابع للأصل.

```
<xsl:template match="ATOM[not(position()=1)]">
  <xsl:value-of select="."/>
</xsl:template>
```

تحدد قاعدة القالب كل عناصر ATOM التي لا تكون أول أو أخر عنصر ATOM التابع للأصل.

```
<xsl:template match =
 "ATOM[not(position()=1 or position()=last())]">
  <xsl:value-of select="."/>
</xsl:template>
```

لا يوجد عامل تشغيل أو خاص ومع ذلك يمكن تكوين واحدة بالاستخدام الحكم لدالـــة not() والكلمات الأساسية and وor وتحدد القاعدة التالية عناصر ATOM التي تكون إما أول أو آخـــر عنصر تابع ولكن ليس الاثنين معاً.

```
<xsl:template
 match="ATOM[(position()=1 or position()=last())
          and not(position()=1 and position()=last())]">
  <xsl:value-of select="."/>
</xsl:template>
```

توجد ثلاثة دوال أخرى تقوم بإرجاع العمليات المنطقية

- دالة true() ترجع دائماً القيمة صحيحة
- دالة false() ترجع دائماً القيمة خطأ
- دالة lang(code) ترجع القيمة صحيحة إذا كانت نقطة البدء الحالية لها نفس اللغة كمـــا هو معطى بسمة xml:lang كما في وسيطة code.

أرقام

تتكون أرقام لغة XSL من ٦٤ بت فاصلة عائمة IEEE ثنائيـــة وتخـــزن الأرقـــام مثـــل ٤٢ أو ٧٠٠٠- التي تبدو مثل الأرقام كلما تطلب الأمر القيم غير الرقمية مثـــل السلاســـل والعمليـــات المنطقية إلى أرقام كلما تطلب الأمر أو بواسطة دالة number() باستخدام القواعد التالية:

- تكون قيمة العمليات المنطقية واحد إذا كانت صحيحة وصفر إذا كانت خطأ.
- يتم إلغاء المسافات البيضاء في مقدمة ونهاية سلسلة ثم تحول إلى رقم بالطريقة المعتـــادة فمثلاً سلسلة "١٢" تحول إلى الرقم ١٢وإذا لم يتم تفسير السلسلة على أنـــها رقـــم يتـــم تحويلها إلى صفر.
- تحويل أجزاء النتائج ومجموعات نقاط البدء إلى سلاسل إلى أرقام.

يوضح المثال التالي لقاعدة تقوم بوضع العناصر التي تلي عنصر اليورانيوم في العدد الـــذري والتي تحدث نتيجة عمليات غير طبيعية وتكون الأعداد الذرية لذلك العناصر أكبر من ٩٢ وهـــو العدد الذري لليورانيوم ويتم تحويل مجموعة نقطة البدء التي نتجت بصـــورة ضمنيـــة بواســـطة ATOMIC_NUMBER إلى قيمة السلسلة لنقطة بدء ATOMIC_NUMBER الحالية وبعـــد ذلـــك تحول تلك السلسلة إلى رقم.

```
<xsl:template match="/PERIODIC_TABLE">
  <HTML>
    <HEAD><TITLE>The TransUranium Elements</TITLE></HEAD>
    <BODY>
```

```
      <xsl:apply-templates select="ATOM[ATOMIC_NUMBER>92]"/>
    </BODY>
  </HTML>
</xsl:template>
```

تقدم لغة XSL عوامل تشغيل العمليات الحسابية الأساسية وهم:

- ♦ (+) للجمع
- ♦ (-) للطرح
- ♦ (*) للضرب
- ♦ div للقسمة وتستخدم علامة الأكثر انتشارا وتدل على القسمة وهي / في أغراض أخرى في لغة XSL.

مثلاً تدرج <xsl:value-of select="2+2"/> السلسلة "٤" في مستند المخرجات وغالباً فتستخدم تلك العمليات كجزء من اختبار وتحدد القاعدة التالية العناصر التي يكون الوزن الذري لهم أكبر من ضعف العدد الذري.

```
<xsl:template match="/PERIODIC_TABLE">
  <HTML>
    <BODY>
      <H1>High Atomic Weight to Atomic Number Ratios</H1>
      <xsl:apply-templates
        select="ATOM[ATOMIC_WEIGHT > 2 * ATOMIC_NUMBER]"/>
    </BODY>
  </HTML>
</xsl:template>
```

يطبع هذا القالب النسبة بين الوزن الذري والعدد الذري

```
<xsl:template match="ATOM">
  <p>
    <xsl:value-of select="NAME"/>
    <xsl:value-of select="ATOMIC_WEIGHT div ATOMIC_NUMBER"/>
  </p>
</xsl:template>
```

تقدم لغة XSL عوامل تشغيل ثنائية ولكنها أقل استخداماً وهم:

- ♦ mod يأخذ باقي رقمين
- ♦ quo يقسم رقمين ثم يختصر الجزء العشري ليحصل على عدد صحيح.

كما يتضمن لغة XSL أربعة دوال تعمل مع الأرقام وهم:

دالة floor() ترجع أكبر عدد صحيح أصغر من الرقم

دالة ceiling() ترجع أصغر عدد صحيح أكبر من الرقم

دالة round() تقرب الرقم إلى أقرب عدد صحيح

دالة sum() ترجع مجموع وسائطه

تقدر قاعدة القالب التالية رقم النيترونات في ذرة عن طريق طـــرح العــدد الــذري وعــدد البروتون من الوزن الذري وهو الوزن المتوسط على التوزيع الطبيعي للنظائر عدد النيترونـــات زائد عدد البروتونات ويلي ذلك تقريب الرقم الناتج أقرب عدد صحيح.

```
<xsl:template match="ATOM">
  <p>
    <xsl:value-of select="NAME"/>
    <xsl:value-of
     select="round(ATOMIC_WEIGHT - ATOMIC_NUMBER)"/>
  </p>
</xsl:template>
```

تحسب تلك القاعدة متوسط الوزن الذري لكل الذرات في الجدول عن طريق جمع كل الأوزان الذرية ثم قسمتها على عدد الذرات.

```
<xsl:template match="/PERIODIC_TABLE">
  <HTML>
    <BODY>
    <H1>Average Atomic Weight</H1>
      <xsl:value-of
       select="sum(from-descendants(ATOMIC_WEIGHT))
             div count(from-descendants(ATOMIC_WEIGHT))"/>
    </BODY>
  </HTML>
</xsl:template>
```

السلاسل

تعتبر السلسلة ترتيب من أحرف Unicode ويمكن تحويل أنواع بيانات أخـــرى إلـــى سلاســل باستخدام دالة string() وفقاً لتلك القواعد:

- تحويل مجموعات نقاط البدء عن طريق وضع قيم نقاط البدء في المجموعة في سلسلة وتحسب قيم نقاط البدء في المجموعة بواسطة عنصر xsl:value-of طبقاً للقواعد المعطاة في جدول ١٤-١.
- تحول إجراء شجرة النتائج بالتعامل معها على أساس أنها محتواه في عنصر واحد ثم تؤخذ قيمة هذا العنصر الافتراضي وبعد ذلك تحسب قيمة هذا العنصر بواسطة عنصر xsl:value-of وفقاً للقواعد المعطاه في جدول ١٤-١ وبهذا يكون نص أجزاء شجرة النتائج وليس الترميز هو الذي وضع في سلسلة.
- يحول رقم إلى سلسلة أرقام تتبع النمط الأوروبي مثل "١٢" أو "٣٠١٤١٥٢٩٢"
- تحول العمليات المنطقية الخطأ إلى كلمة اللغة الإنجليزية خطأ "false" و تحول العمليات المنطقية الصحيحة إلى كلمة اللغة الإنجليزية صحيحة "true".
- تحتوي لغة XSL على سبع دوال أخرى بخلاف ()string تعمل السلاسل ويلخصهم جدول ١٤-٥.

الجدول ١٤-٥
الدوال التي تعمل على السلاسل

الدالة	نوع الإرجاع	ترجع
starts-with(main_string prefix_string)	عمليات منطقية	يكون الإرجاع صحيح إذا بدأت main_string بتعليمات prefix_string وتكون خطأ في أي حالة أخرى
Contains(containing_string, contained_string)	عمليات منطقية	تكون صحيحة إذا كانت contained_string جزء من contained_string قد تكون خطأ في أي حالة أخرى
substring-before(string, marker-string)	سلسلة	ترجع جزء السلسلة من بدايتها إلى أول حدوث لعلامة السلسلة marker-string ولكنه لا يشتمل عليها
substring-after(string, marker-string)	سلسلة	ترجع جزء السلسلة من نهاية أول جدول لعلامة السلسة إلى نهاية السلسلة

الجدول ١٤–٥

الدوال التي تعمل على السلاسل

الدالة	نوع الإرجاع	ترجع
normalize(string)	سلسلة	ترجع string بعد إزالة المسافات البيضاء السابقة واللاحقة واستبدال المسافات البيضاء المتعددة بمسافة واحدة وإذا حذفت الوسيطة يتم تسوية قيمة السلسلة لنقطة بدء السياق
translate(string, replaced_text, replacement_text)	سلسلة	ترجع string مع حدوث أحرف في replaced_text تستبدل بواسطة الأحرف المتوافقة معها من replaced_text
concat(string1, string2, ...)	سلسلة	ترجع سلاسل لأي عدد من السلاسل يتم تمريره لوسيطه في الترتيب الذي تم تمريرهم به
format-number(number, string format-string, locale-string)	سلسلة	ترجع نموذج السلسلة لرقم منسق طبقاً format-string المحدد في الموقع المعين بواسطة locale-string كما في فئة Java 1.1 وهي java.text. DecimalFormat ولهذا انظر الموقع التالي: http: //java.sun.com/products/ jdk/1.1/docs/api/java. text.DecimalFormat.html

أجزاء شجرة النتائج

يعتبر جزء شجرة جزء من مستند XML وهو ليس نقطة بدء كاملة أو مجموعة نقاط بدء فعلى سبيل المثال فإن استخدام دالة doc() مع URI يشير إلى وسط عنصر قد ينتج عنه جزء شجرة نتيجة وترجع أجزاء شجرة النتائج بواسطة بعض الدوال الملحقة وهي دوال خاصة بتطبيق وتثبيت لغة XSL.

لا تفيد أجزاء شجرة النتائج كثيراً لأنها ليست مستند XML صحيح التكوين والعمليات الوحيدة التي تتيحها هي تحويلهم إلى سلسلة أو عمليات منطقية باستخدام ()string و()boolean على التوالي.

قواعد القوالب الافتراضية

تسبب الدقة المتناهية المطلوبة لرسم هيكلة مستند XML في ورقة نمط XSL بعض الضيق خاصة إذا كان المستند لا يتيح ترتيب ثابت ومتوقع مثل الجدول الدوري ولكن مثل كثير من صفحات ويب يلقي بالعناصر معاً دون ترتيب ففي تلك الحالات يجب وجود قواعد عامة لتمد بها العنصر وتطبق القالب عليه دون الاهتمام بمكان ظهوره في مستند المصدر.

تعرف لغة XSL قاعدتي قالب افتراضية بطريقة ضمنية في أوراق النمط لجعل تلك العملية أسهل وتدرج القاعدة الافتراضية الأولى بصورة متصلة بشجرة العنصر وتطبق قوالب على توابع العناصر ويضمن هذا أن كل قواعد القوالب التي تنطبق على العناصر يتم تطبيقها وتطبق القاعدة الافتراضية الثانية على نقاط بدء النص وتنسخ قيمتهم إلى سياق المخرجات وتعني القاعدتين معاً أن ورقة نمط XSL فارغة بدون عناصر تنتج بيانات الأحرف الأساسية لمستند XML كمخرجات.

القاعدة الافتراضية للعناصر

تطبق القاعدة الافتراضية الأولى على نقاط بدء العنصر لأي نوع أو على نقطة بدء الجذر.

```
<xsl:template match="*|/">
  <xsl:apply-templates/>
</xsl:template>
```

تعتبر العلامة /|* هي اختصار لغة XSL لجملة أي عنصر أو نقطة بدء جذر والهدف من هذه القاعدة هو تأكيد أن كل العناصر تعالج بصورة متتالية حتى في حالة عدم الوصول إليهم باتباع القواعد الصريحة وهذا يعني أنه طالما لم يتم تجاوز تلك القاعدة بأخرى "خاصة بالنسبة لعنصر الجذر" تتم معالجة كل نقاط بدء العنصر.

في حالة وجود أي قاعدة واضحة لأصل أي عنصر فلن يتم تنشيط هذه القاعدة بالنسبة للعناصر التابعة إلا إذا كانت قاعدة القالب للأصل لهما تابع xsl:apply-templates ومثال لذلك يمكن وقف كافة عمليات المعالجة بواسطة مطابقة عنصر الجذر وعدم تطبيق القوالب أو استخدام xsl:for-each لمعالجة التوابع كما يلي:

```
<xsl:template match="/">
</xsl:template>
```

القاعدة الافتراضية لنقاط النص

قامت عدة أمثلة بوضع محتويات بعض العناصر كمخرجات دون أخذ قيمة العنصر الذي يتم إخراجه فعلياً وقد تم تزويد هذا المحتوى بواسطة قاعدة XSL الافتراضية لنقاط بدء النص التي تقع كمحتوى عنصر وتلك القاعدة هي:

```
<xsl:template match="text()">
  <xsl:value-of select="."/>
</xsl:template>
```

تطابق تلك القاعدة كل نقاط بدء النص (match="text()") وتقوم بوضع قيمة نقطة بدء النص كمخرج (<xsl:value-of select="."/>) وبمعنى أخر فهي تنسخ النص من الإدخال إلى المخرجات.

تضمن هذه القاعدة أنه على أقل تقدير يكون نص العنصر مخرج حتى في حالة عدم تطابق قاعدة مع العنصر بصورة محددة وقد تتجاوز قاعدة أخرى عن هذه القاعدة لبعض العناصر المحددة التي تريد فيها إما أقل أو أكبر من محتوى نص عنصر.

معنى القاعدتين الافتراضيتين

تشير القاعدتين الافتراضيتين معاً إلى أن تطبيق ورقة نمط فارغة بها فقط عنصر xsl:stylesheet وليس بها عناصر تابعة مثل تعليمات برمجة ١٤-١٤ على مستند XML يجعل ورقة النمط تنتج كل #PCDATA من العناصر الموجودة في الإدخال إلى المخرجات وعامة لا ينتج عن تلك الطريقة ترميز وتعتبر تلك القواعد ليست ذات أهمية كبيرة وبالتالي فإن أي مطابقة أخرى لها الأولوية على هاتين القاعدتين.

تعليمات البرمجة ١٤-١٤ : ورقة نمط XML فارغة

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

</xsl:stylesheet>
```



إحدى أهم الأوجه التي تسبب الاختلاط بشأن لغة XSL في Internet Explorer 5.0 هو أنها لا تقدم أي من هاتين القاعدتين الافتراضيتين ويجب التأكد من مطابقة أي نقطة بدء تريد المطور وضع محتوياتها بما في ذلك مشتقات نقطة البدء في مستند المخرجات بطريقة واضحة.

تحديد المخرجات المضمنة

يجب تأجيل أي قرار بشأن الترميز الواجب التعبير عنه إلى أن تتم قراءة مستند الإدخال فعلى سبيل المثال قد يرغب المطور في تعبير محتويات عنصر FILENAME إلى سمة HREF لعنصر A أو استبدال نوع واحد في الإدخال بأنواع عناصر مختلفة متعددة في المخرجات بناء على قيمة واحد من سماته ويتم تحقيق ذلك باستخدام عناصر xsl:element, xsl:attribute وxsl:pi وxsl:comment وxsl:text elements وتستخدم تعليمات لغة XSL في محتويات تلك العناصر وتستخدم قوالب قيمة السمة في قيم سمات تلك العناصر لتغيير تلك المخرجات.

استخدام قوالب قيم السمات

تنتج قوالب قيم السمات من محتوى العنصر في الإدخال إلى قيم السمات في ورقة النمط ومن ورقة النمط يمكن كتابتها في المخرجات ولنفترض كمثال رغبة المطور في تحويل الجدول الدوري إلى عناصر ATOM فارغة باستخدام النموذج التالي الذي يعتمد على السمة.

```
<ATOM NAME="Vanadium"
  ATOMIC_WEIGHT="50.9415"
  ATOMIC_NUMBER="23"
  OXIDATION_STATES="5, 4, 3, 2"
  BOILING_POINT="3650K"
  MELTING_POINT="2163K"
  SYMBOL="V"
  DENSITY="6.11 grams/cubic centimeter"
/>
```

يجب لفعل ذلك استخراج محتويات العناصر في مستند الإدخال ووضع العناصر في قيم السمات في مستند المخرجات وتبدو أول تعليمات برمجة كما يلي:

```
<xsl:template match="ATOM">
   <ATOM NAME="<xsl:value-of select='NAME'/>"
```

```
  ATOMIC_WEIGHT="<xsl:value-of select='ATOMIC_WEIGHT'/>"
  ATOMIC_NUMBER="<xsl:value-of select='ATOMIC_NUMBER'/>"
/>
```

تعتبر </xsl:template> مستند XML غير صحيح التكوين ولأنه لا يمكن استخدام علامة (>) داخل سمة بالإضافة إلى صعوبة كتابة برنامج يستطيع توزيع ذلك في الحالات الأكثر شيوعاً.

بدلاً من ذلك قيم السمات الداخلية تأخذ البيانات المضمنة في الأقواس الحاصرة وهي { } مكان عنصر xsl:value-of وبذلك تكون الطريقة الصحيحة لكتابة ما سبق هو ما يلي.

```
<xsl:template match="ATOM">
 <ATOM NAME="{NAME}/>"
  ATOMIC_WEIGHT="{ATOMIC_WEIGHT}/>"
  ATOMIC_NUMBER="{ATOMIC_NUMBER}/>"
 />
</xsl:template>
```

في مستند المخرجات تستبدل {NAME} بقيمة عنصر تابع NAME لنقطة البدء الحالية وتستبدل {ATOMIC_WEIGHT} بقيمة عنصر ATOMIC_WEIGHT التابع لنقطة البدء الحالية وتستبدل {ATOMIC_NUMBER} بقيمة عنصر ATOMIC_NUMBER التابع وهكذا.

تكون قوالب قيمة السمات نماذج أكثر تعقيداً عن اسم العنصر ويمكن استخدام أي من تعبيرات السلسلة التي تم مناقشتها فيما سبق في قالب قيمة السمة فعلى سبيل المثال تحدد قاعدة القالب عناصر DENSITY في النموذج المستخدم في تعليمات برمجة ١٤-١.

```
<xsl:template match="DENSITY">
 <BULK_PROPERTY
  NAME="DENSITY"
  ATOM="{../NAME}"
  VALUE="{.}"
  UNITS="{@UNITS}"
 />
</xsl:template>
```

تحول قاعدة القالب العناصر إلى عناصر BULK_PROPERTY وتظهر كما يلي:

```
<BULK_PROPERTY NAME="DENSITY" ATOM="Helium" VALUE="
  0.1785
 " UNITS="grams/cubic centimeter"/>
```

لا تفيد قيم السمات لقالب قيمة سمة واحدة ويمكن دمج قالب قيمة سمة بالبيانات الحرفيـــــة أو بقوالب قيم سمات أخرى فعلى سبيل المثال تطابق قاعدة القالب عناصر ATOM وتستبدلها بالاسم المنسق كارتباط لملف في تنسيق H.html وHe.html ... الخ. يشتق اسم الملف من قالب قيمـــة السمة {SYMBOL} بينما تقدم البيانات الحرفية الملحق والنقط.

```
<xsl:template match="ATOM">
  <A HREF="{SYMBOL}.html">
    <xsl:value-of select="NAME"/>
  </A>
</xsl:template>
```

يمكن تضمين أكثر من قالب قيمة سمة واحد في قيمة السمة ولهذا تتضمن قاعدة القالب التالية وحدات الكثافة كجزء من سمة VALUE بدلاً من جعلها سمة منفصلة.

```
<xsl:template match="DENSITY">
  <BULK_PROPERTY
   NAME="DENSITY"
   ATOM="{../NAME}"
   VALUE="{.} {@UNITS}"
  />
</xsl:template>
```

تستخدم قوالب قيم السمات في قيم معظم السمات في ورقة نمط XSL ويعتبر هذا مهماً بصفـــة خاصة في عناصر xsl:element وxsl:attribute وxsl:pi elements بينما تتيح قوالب قيـــم السمات للمصمم تأجيل قراره بشأن أي العناصر أو السمات أو تعليمات المعالجة يجب ظـــهورها في المخرجات إلى أن يتم الانتهاء من قراءة مستند الإدخال ولا يمكن استخدام قوالب قيم السـمات كقيمة كل من سمة match أو سمة select أو سمة xmlns أو سمة تقدم اسم عنصر تعليمـــات XSL أخر أو سمة لعنصر عالي القيمة وهو عنصر يكون تابع مباشر لعنصر xsl:stylesheet.

المرجع يناقش الفصل ١٨ "Namespace" سمات xmlns.

إدراج عناصر في المخرجات باستخدام xsl:element

تدرج العناصر في مستند المخرجات باستخدام العناصر الحرفية نفسها فعلى سبيل المثـــال لإدراج العنصر أ تكتب كلاً من <P> و</P> في أماكنهم المناسبة في ورقة النمط وفي بعض الأحيـــان يجب استخدام التفاصيل من مستند الإدخال لتحديد أي العناصر يتم وضعها في المخرجات وقـــد

يحدث هذا عند القيام بالتمويل من مرادف المصدر الذي يستخدم السمات للمعلومات لمرادف المخرجات الذي يستخدم عناصر لنفس المعلومات.

يدرج عنصر xsl:element عنصر في مستند المخرجات ويعطي اسم العنصر بواسطة قالب قيمة سمة في سمة name الخاصة بالعنصر xsl: element وتشتق محتويات العنصر من محتويات عنصر xsl:element والتي قد يتضمن xsl:attribute وxsl:pi وxsl:comment والتي سيتم مناقشتها فيما يلي لإدراج تلك العناصر.

لنفترض مثلاً أن المطور يرغب في استبدال عناصر ATOM بالعناصر التالية GAS وLIQUID وSOLID وذلك بناء على قيمة سمة STATE وباستخدام عنصر xsl:element يمكن لقاعدة واحدة أن تؤدي تلك المهمة بواسطة تحويل قيمة سمة STATE لاسم العنصر وما يلي يوضح ذلك:

```
<xsl:template match="ATOM">
  <xsl:element name="{@STATE}">
    <NAME><xsl:value-of select="NAME"/></NAME>
    <!- rules for other children ->
  </xsl:element>
</xsl:template>
```

يمكن أداء معظم الحسابات المطلوبة باستخدام قوالب قيم سمات أكثر تطوراً.

إدراج سمات في مستند المخرجات باستخدام xsl:attribute

يمكن تضمين السمات في مستند المخرجات باستخدام السمات الحرفية فعلى سبيل المثال ليتم إدراج عنصر DIV مع سمة ALIGN والتي تحمل قيمة CENTER تتم كتابة <DIV ALIGN="CENTER"> و</DIV> في أماكنها الصحيحة في ورقة النمط وفي أغلب الأحيان يتم الاعتماد على بيانات تم قراءتها من مستند الإدخال لتحديد قيمة السمة وفي بعض الأحيان تحديد اسم السمة.

لتفترض مثلاً أنك ترغب في ورقة نمط تحدد أسماء الذرات وتقوم بتنسيقهم كارتباطات لملفات H.html وHe.html وLi.html وهكذا.

```
<LI><A HREF="H.html">Hydrogen</A></LI>
<LI><A HREF="He.html">Helium</A></LI>
<LI><A HREF="Li.html">Lithium</A></LI>
```

يحصل كل عنصر مختلف في مستند الإدخال على قيمة مختلفة لسمة HREF ويحسب عنصر xsl:attribute قيمة واسم السمة وتقوم بإدراجها في مستند المخرجات ويعتبر كل عنصر xsl:attribute تابع إما لعنصر xsl:element أو لعنصر حرفي والسمة التي تحتسب بواسطة xsl:attribute يتم إلحاقها بالعنصر الذي تم حسابه بواسطة أصله في مستند المخرجات ويحدد اسم السمة بواسطة سمة name لعنصر xsl:attribute وتعطي قيمة السمة بواسطة محتويات عنصر xsl:attribute وتستخدم قاعدة القالب الثانية في إنتاج المخرجات السابقة.

```
<xsl:template match="ATOM">
  <LI><A>
    <xsl:attribute name="HREF">
      <xsl:value-of select="SYMBOL"/>.html
    </xsl:attribute>
    <xsl:value-of select="NAME"/>
  </A></LI>
</xsl:template>
```

تأتي عناصر xsl:attribute قبل أن تحتوي أخر لعنصر الأصل الخاص بهم لا يمكن إضافة سمة لعنصر بعد بدء كتابة محتوياته ويعتبر القالب التالي غير مسموح به لهذا السبب.

```
<xsl:template match="ATOM">
  <LI><A>
    <xsl:value-of select="NAME"/>
    <xsl:attribute name="HREF">
      <xsl:value-of select="SYMBOL"/>.html
    </xsl:attribute>
  </A></LI>
</xsl:template>
```

تعريف مجموعات السمات

تحتاج معظم الأحيان إلى تطبيق نفس مجموعة السمات لعناصر مختلفة تكون من نفس الفئة أو من فئات مختلفة فمثلاً قد يرغب المطور في تطبيق سمة style لكل خلية في جدول HTML ولتبسيط ذلك الإجراء يمكن تعريف سمة أو أكثر كأعضاء مجموعة سمة في أعلى مستويات ورقة النمط باستخدام xsl:attribute-set ثم قم بتضمين مجموعة السمة في عنصر باستخدام xsl:use.

يعرف عنصر xsl:attribute-set عنصر يسمى cellstyle له سمة font-family تتكـــون من سمات New York وTimes New Roman وTimes, serif وfont-size ويكـــون حجـم الخط ١٢ نقطة.

```
<xsl:attribute-set name="cellstyle">
  <xsl:attribute name="font-family">
    New York, Times New Roman, Times, serif
  </xsl:attribute>
  <xsl:attribute name="font-size">12pt</xsl:attribute>
</xsl:attribute-set>
```

تطبق قاعدة القالب تلك السمات على عناصر إما في مستند على عنـــاصر td فـــي مســتند المخرجات فكما مع xsl:attribute يجب وضع عنصر xsl:use الذي يدرج مجموعة السمة قبل أي محتوى يتم إضافته كتابع لعنصر إما.

```
<xsl:template match="ATOM">
  <tr>
    <td>
      <xsl:use attribute-set="cellstyle"/>
      <xsl:value-of select="NAME"/>
    </td>
    <td>
      <xsl:use attribute-set="cellstyle"/>
      <xsl:value-of select="ATOMIC_NUMBER"/>
    </td>
  </tr>
</xsl:template>
```

في حالة استخدام عنصر لأكثر من مجموعة سمة يتم تطبيق كل السمات من كل المجموعـات على العنصر وإذا قامت أكثر من مجموعة سمة واحدة بتعريف نفس السمة بقيـــم مختلفـــة يتـــم استخدام السمة التي توجد في أكثر المجموعات أهمية وتعتبر ورقة النمط التي تقوم مجموعـــات سمات بنفس الأهمية بتعريف نفس السمة خطأ.

إنشاء تعليمات المعالجة باستخدام xsl:pi

يضع عنصر xsl:pi تعليمات معالجة في مستند المخرجات ويحدد الهدف من تعليمات المعالجـــة بواسطة سمة name مطلوبة وتصبح محتويات عنصر xsl:pi هي محتويات تعليمات المعالجـــة ومثال لذلك القاعدة التالية حيث تستبدل عناصر PROGRAM بتعليمات معالجة gcc.

```
<xsl:template select="PROGRAM">
  <xsl:pi name="gcc"> -O4</xsl:pi>
</xsl:template>
```

تستبدل عناصر PROGRAM في مستند الإدخال تعليمـــات المعالجـــة التاليـــة فـــي مســـتند المخرجات.

```
<?gcc -O4?>
```

يتضمن محتويات عنصر xsl:pi عنـــاصر xsl:value وعنـــاصر xsl:apply-templates على شريط أن تكون نتائج تلك التعليمات نص فقط وما يلي يوضح ذلك:

```
<xsl:template select="PROGRAM">
  <xsl:pi name="gcc">-O4 <xsl:value-of select="NAME"/></xsl:pi>
</xsl:template>
```

أحد الاستخدامات الشائعة لعنصر xsl:pi هو إدراج تعريف XML عند إنتاج مستند XML فمن مستند XML أخر على الرغم من أن تعريف XML من الناحية النفسية ليس تعليمات معالجة مثلاً.

```
<xsl:pi name="xml">version="1.0" standalone="yes"</xsl:pi>
```

قد لا يتضمن عنصر xsl:pi تعليمات معالجة xsl:element وتعليمات أخرى تنتج عنـــاصر وسمات في النتيجة بالإضافة إلى ذلك قد لا يتضمن عنصر xsl:pi أي تعليمات أو نص حرفـــي تقوم بإدراج علامة في مستند المخرجات حيث تقوم تلك العلامات بإنشاء تعليمات المعالجة قبـــل وقتها المحدد.

إنشاء تعليقات باستخدام xsl:comment

يدرج عنصر xsl:comment تعليق في مستند المخرجات مع العلم أنه لـــيس لـــه ســـمات وأن المحتويات هي نص التعليق ومثال على ذلك:

```
<xsl:template select="ATOM">
  <xsl:comment>There was an atom here once.</xsl:comment>
</xsl:template>
```

تستبدل تلك القاعدة نقاط بدء ATOM بالمخرجات التالية:

```
<!-There was an atom here once.->
```

تتضمن محتويات عنصر xsl:comment عنـــاصر xsl:value-of وعنـــاصر xsl:apply-templates على شرط أن تكون نتائج تلك التعليمات هي نص فقـــط وقـــد لا تحتـــوي علـــى xsl:element أو تعليمات أخرى تنتج عناصر وسمات في النتيجة وبالإضافة إلى ذلـــك قـــد لا تتضمن xsl:comment على أي تعليمات أو نص حرفي يدرج فاصل مزدوج في التعليق وينتـــج عن هذا تعليق غير صحيح التكوين في مستند المخرجات وهذا ما لا يسمح به.

إنشاء نص باستخدام xsl:text

يدرج عنصر xsl:text محتويات في مستند المخرجات كنص حرفي فعلى سبيل المثال تستبدل القاعدة التالية كل عنصر ATOM بسلسلة كان "هناك ذرة هنا فيما سبق"There was an atom here once".

```
<xsl:template select="ATOM">
  <xsl:text>There was an atom here once.</xsl:text>
</xsl:template>
```

لا يستخدم عنصر xsl:text بكثرة لأن في معظم الأحيان يكون من الأسهل كتابة النص ومع ذلك فلعنصر xsl:text ميزة واحدة وهي أنه يحفظ المسافات البيضاء بالتحديد ويكون هذا مفيداً عند التعامل مع الشعر وتعليمات مصدر جهاز الكمبيوتر البرمجية أو أي معلومات أخرى تكون المسافات البيضاء هامة فيه.

نسخ نقطة البدء الحالية باستخدام xsl:copy

ينسخ عنصر xsl:copy نقطة بدء المصدر إلى المخرجات ولا يتم نسخ العناصر الفرعية والسمات والمحتويات الأخرى بطريقة أكبر وعموماً فإن محتويات عنصر xsl:copy هي عناصر xsl:template الذي يحد تلك الأشياء لنسخها أيضاً ويكون هذا مفيداً عند تحويل مستند من مفردات لغة ترميز إلى مفردات مماثلة أو إلى مفردات لغة ترميز مرتبطة بها إلى حد كبير ومثال ذلك ترسل قاعدة القالب السمات والعناصر التابعة من الذرة وتستبدلها بقيمة محتوياتها.

```
<xsl:template match="ATOM">
  <xsl:copy>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>
```

يمكن الاستفادة من قالب xsl:copy في أنه يتيح التمويل التعريفي وهذا يعني تحويل من مستند إلى نفسه ويبدو مثل هذا التمويل كما يلي:

```
<xsl:template match="*|@*|comment()|pi()|text()">
  <xsl:copy>
    <xsl:apply-templates select="*|@*|comment()|pi()|text()"/>
  </xsl:copy>
</xsl:template>
```

يتم ضبط التمويل التعريفي لإنتاج مستندات مماثلة ومثال لذلك فإن تعليمات برمجـــة ١٤–١٥ عبارة عن ورقة نمط تزيل التعليقات من المستند ويترك باقي المستند دون المساس به وهذا نتيجة ترك نقطة بدء comment() خارج قيم سمات match وselect في التمويل التعريفي.

تعليمات البرمجة ١٤–١٥: ورقة نمط XSL تزيل التعليقات من المستند

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="*|@*|pi()|text()">
    <xsl:copy>
      <xsl:apply-templates select="*|@*|pi()|text()"/>
    </xsl:copy>
  </xsl:template>

</xsl:stylesheet>
```

تنسخ xsl:copy نقاط بدء المصدر فقط ويمكن نسخ نقاط بدء أخرى أو من الممكن نسخ أكثر من واحدة باستخدام xsl:copy-of ومثال على ذلك تقدم تعليمات برمجة ١٤–١٦ ورقـــة نمـط تستخدم xsl:copy-of لإزالة العناصر التي ليس لها نقطة انصهار من الجـدول الـدوري عـن طريق نسخ عناصر ATOM فقط التي لها عناصر تابعة هي MELTING_POINT.

تعليمات البرمجة ١٤–١٦: ورقة نمط تنسخ فقط عناصر ATOM التي لها عناصر تابعة هي MELTING_POINT

```
<?xml version="1.0""?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

   <xsl:template match="/PERIODIC_TABLE">
     <PERIODIC_TABLE>
       <xsl:apply-templates select="ATOM"/>
```

```
 </PERIODIC_TABLE>
</xsl:template>

<xsl:template match="ATOM">
  <xsl:apply-templates
   select="MELTING_POINT"/>
</xsl:template>

<xsl:template match="MELTING_POINT">
  <xsl:copy-of select="..">
    <xsl:apply-templates select="*|@*|pi()|text()"/>
  </xsl:copy-of>
</xsl:template>

<xsl:template match="*|@*|pi()|text()">
  <xsl:copy>
    <xsl:apply-templates select="*|@*|pi()|text()"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```



ما سبق هو مثال للغة تحويل XSL من مفردات المصدر إلى نفس المفردات وعلى عكس معظم الأمثلة في هذا الفصل أنها لا تحول إلى لغة HTML صحيحة التكوين.

حساب نقاط البدء باستخدام xsl:number

يدرج عنصر xsl:number عدد صحيح منسق في مستند المخرجات وتعطي قيمة الرقم الصحيح عن طريق تقريب الرقم المحسوب بواسطة سمة expr إلى أقرب رقم صحيح تم تنسيقه وفقاً لقيمة سمة format ويتم تقديم افتراضات معقولة لكل من السمتين ومثال لذلك انظر إلى ورقة النمط لعناصر ATOM في تعليمات برمجة ١٤-١٧.

تعليمات البرمجة ١٤-١٧: ورقة نمط XSL تقوم بحساب الذرات

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="PERIODIC_TABLE">
    <html>
      <head><title>The Elements</title></head>
      <body>
        <table>
          <xsl:apply-templates select="ATOM"/>
        </table>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="ATOM">
    <tr>
      <td><xsl:number expr="position()"/></td>
      <td><xsl:value-of select="NAME"/></td>
    </tr>
  </xsl:template>

</xsl:stylesheet>
```

عند تطبيق ورقة النمط السابقة على تعليمات برمجة ١٤-١ تظهر المخرجات التالية:

```
<html><head><title>The
Elements</title></head><body><table><tr><td>1</td><td>Hydrog
en</td></tr>
<tr><td>2</td><td>Helium</td></tr>
</table></body></html>
```

يحصل الهيدروجين على رقم ١ لأنه هو عنصر ATOM الأول في أصله ويحصل الهليوم على رقم ٢ لأنه عنصر ATOM الثاني في أصله وكون تلك الأرقام هي الأعداد الذرية لكل من الهيدروجين والهليوم ليس له تأثير أساسي في أن تعليمات برمجة ١٤-١ تم ترتيبها حسب ترتيب العدد الذري.

الأعداد الافتراضية

كل ما تحتاجه هو استخدام سمة expr لحساب الأعداد ومع ذلك إذا تم حذف سمة expr يتم استخدام موقع نقطة البدء الحالية في شجرة المصدر على أنه الرقم ومع ذلك يمكن ضبط هذا الافتراضي باستخدام السمات الثلاثة التالية:

- level
- count
- from



تلك السمات الثلاثة هم متواجدين من نسخ تمهيدية سابقة من XSL لم تدعم التعبيرات الأكثر تعقيداً المتاحة حالياً وإما تسبب ذلك في اختلاط الأمر على المستخدم يمكن تجاهلهم واستخدام expr بدلاً منهم.

سمة Attribute

افتراضياً وبدون وجود سمة expr يقوم عنصر xsl:number بحساب نقاط بدء المرتبطة بنقطة بدء المصدر فمثلاً إذا كانت عناصر ATOMIC_NUMBER مرقمة بدلاً من عناصر ATOM لن يحصل أي من العناصر على رقم أعلى من واحد لأن عنصر ATOM لا يحصل إلا على واحد من عناصر ATOMIC_NUMBER التابعة وعلى الرغم من أن المستند يحتوي على أكثر من عنصر ATOMIC_NUMBER واحد فتلك العناصر ليست متصلة.

يتسبب إعداد سمة level لعنصر xsl:number على any يقوم بحساب كل العناصر من نفس النوع كنقطة البدء الحالية في المستند وهذا لا يتضمن فقط العناصر التي تطابق القاعدة الحالية ولكن كل العناصر من النوع الصحيح وفي حالة تحديد العدد الذري للغازات فقط مثلاً فالسؤال والعناصر الصلبة سيتم حسابها أيضاً حتى وإن لم يظهروا في مستند المخرجات والقواعد التالية توضح ما سبق.

```
<xsl:template match="ATOM">
  <xsl:apply-templates select="NAME"/>
</xsl:template>

<xsl:template match="NAME">
  <td><xsl:number level="any"/></td>
  <td><xsl:value-of select="."/></td>
</xsl:template>
```

حيث أن إعداد level على any فإن المخرجات لا تبدأ من رقم ١ مع كـل عنصـر NAME جديد وما يلي يوضح ذلك:

```
<td>1</td><td>Hydrogen</td>

<td>2</td><td>Helium</td>
```

تبدو المخرجات في حالة إلغاء سمة level أو إعدادها على القيمة الافتراضية وهـي single كما يلي:

```
<td>1</td><td>Hydrogen</td>

<td>1</td><td>Helium</td>
```

يعين خيار أخر أقل فائدة عن السابق سمة level لعنصر xsl:number على multi لتحديـد أن كلا التوابع لنقطة البدء الحالية وأصلها، وليس العناصر التابعة التي لا تكون توابع لنقطة البدء الحالية، يجب حسابها.

سمة count

افتراضياً وبدون وجود سمة expr يتم حساب العناصر من نفس نوع عنصر نقطة البدء الحاليـة فقط ومع ذلك يمكن إعداد سمة count لعنصر xsl: number لتعبير محدد لتعيين ما الذي سـيتم حسابه والقاعدة التالية تطبق العدد على كل العناصر التابعة لعنصر ATOM.

```
<xsl:template match="ATOM/*">
  <td><xsl:number count="*"/></td>
  <td><xsl:value-of select="."/></td>
</xsl:template>
```

تبدو مخرجات تطبيق تلك القاعدة كما يلي:

```
<td>1</td><td>Helium</td>
<td>2</td><td>He</td>
<td>3</td><td>2</td>
<td>4</td><td>4.0026</td>
<td>5</td><td>1</td>
<td>6</td><td>4.216</td>
<td>7</td><td>0.95</td>
<td>8</td><td>
  0.1785
</td>
```

سمة from

محتوى سمة from على تعبير select الذي يحدد العنصر الذي يتم بدء العد بـــه فـــي شـــجرة الإدخال ومع ذلك يبدأ العد من واحد وليس اثنين أو عشرة أو غير ذلـــك وتغيـــير ســـمة from للعنصر المفترض أن يكون العنصر الأول.

رقم لتحويل السلسلة

تم افتراض إلى تلك النقطة في الكتاب أن الأرقام تبدو مثل الترقيم العادي ١، ٢، ٣ ...الخ وهـــو الترقيم الأوروبي الذي يبدأ بالرقم واحد ويراد في كل مرة رقم إلا أن ذلك ليس هو الترقيم الوحيد المتاح فيمكن مثلاً لصفحات المقدمة في كتاب أن تبدأ بـــالترقيم الرومـــاني الصغـــير ١، ٢، ٣، ....الخ وتستخدم دول مختلفة مصطلحات مختلفة لتجميع الأرقام وفصل الأرقام الصحيحة وإجراء الكسور العشرية عن الرقم الأصلي وتمثيل رموز الأرقام المختلفة وكل ذلك يمكن ضبطـــه عـــن طريق خمس سمات لعنصر xsl:number وهم:

- format
- letter-value
- digit-group-sep
- n-digits-per-group
- sequence-src

سمة format

يمكن ضبط نمط الترقيم المستخدم بواسطة xsl:number باستخدام سمة format وتكون قيمة تلك السمة أحد القيم التالية:

- i: تنتج ترتيب الترقيم الروماني ذو الأحرف الصغيرة i, ii. iii, iv, v, vi.
- I: تنتج ترتيب الترقيم الروماني والأحرف الكبيرة I, II, III, IV, V, VI.
- a: تنتج ترتيب الأحرف اللاتينية الصغيرة a, b, c, d, e, f.
- A: تنتج ترتيب الأحرف اللاتينية الكبيرة A, B, C, D, E, F

توضح القاعدة التالية ترقيم الذرات بالترقيم الروماني ذو الأحرف الكبيرة.

```
<xsl:template match="ATOM">
  <P>
    <xsl:number expr=سposition()" format="I"/>
```

```
      <xsl:value-of select="."/>
    </P>
  </xsl:template>
```

يمكن ضبط الرقم أو الحرف الذي يبدأ الترقيم به عن طريق تغيير قيمة سمة format فمثـــلاً ليبدأ الترقيم عند الرقم ٥ يتم ضبط ســـمة "5" =format وليبـــدأ الـــترقيم عنـــد iii يتـــم ضبـــط format="iii".

يمكن تحديد الترقيم العشري باستخدام الأصفار Leading zeroes عن طريق تضمين أرقـــام Leading zeroes في سمة format فمثلاً إعداد "format="01 ينتج عنه الترتيب 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12 ويكون ذلك مفيداً عند وضع الأرقام في صفـوف الأعمدة.

سمة letter-value

تميز سمة letter-value بين الحروف التي تعامل على أساس أنها أرقام والحروف التي تعـــامل على أساس أنها حروف فمثلاً في حالة استخدام "format="I لبدء الترتيب من I, J, K, L, M, N بدلاً من I, II, III, IV, V, VI, يجب ضبط سمة letter-value على alphabetic وضبـــط سمة letter-value على other يعني الترتيب الرقمي وما يلي يوضح ذلك.

```
<xsl:template match="ATOM">
  <P>
   <xsl:number expr="position()"
           format="I" letter-value="alphabetic"/>
   <xsl:value-of select="."/>
  </P>
</xsl:template>
```

سمات Group Separator

يتجه الناس في الولايات المتحدة إلى كتابة الأرقام الكبيرة باستخدام الفاصلة لتجمع كـــل ثلاثـــة أرقام معاً مثل ٤,٥٦٧,٣٠٢ وتستخدم العديد من اللغـــات والـــدول النقطـــة والمســـافة لتفضـــل المجموعات بدلاً من ذلك فتكتب مثلاً ٤.٥٦٧.٣٠٢ أو ٤ ٥٦٧ ٣٠٢ وفي العديد من الدول مـــن المعتاد تجميع الأرقام الكبيرة كل ٤ أرقام بدلاً من ٣ أرقام فمثلاً ٤,٥٦٧٣ وفي حالة التعامل مـــع قائمة كبيرة قد يحتوي على ألف أو اكثر من تلك العناصر يجب الاهتمام بتلك النقطة.

تحدد سمة digit-group-sep فاصل المجموعات المستخدم بين مجموعات الأرقـــام وتحـــدد سمة n-digits-per-group عدد الأرقام المستخدمة في مجموعة وعلمه يمكن جعل تلك السمات contingent في اللغة وما يلي مثال على ذلك:

```
<xsl:number digit-group-sep=" "/>
```

سمة sequence-src

في حالة أراء المطور استخدام ترتيب غير اعتيادي مثل قائمة بسلسلة تاريخ مثـل ١-١-١٩٩٩، ١-٢-١٩٩٩، ١-٣-١٩٩٩ ...الخ أو قائمة بمضاعفات رقم عشـرة مثـل ١٠، ٢٠، ٣٠، ٤٠، يمكن تخزين هذه القائمة وفصلها بواسطة مسافات بيضاء فـي مسـتند منفصـل وقيمـة سـمة sequence-src تمثل URL المطلق أو نسبي لهذا المستند ومثال على ذلك:

```
<xsl:number sequence-src="1999.txt"/>
```

فرز عناصر المخرجات

يفرز عنصر xsl:sort عناصر المخرجات إلى ترتيب يختلف عن ترتيب ظهورها في الإدخـال ويظهر عنصر xsl:sort لتـابع لعنصـر xsl:apply-templates أو عنصـر xsl:for-each وتعرف سمة select لعنصر xsl: sort element الأساس المستخدم لفرز مخرجات العنـاصر نتيجة استخدام عنصر xsl:apply-templates أو عنصر xsl:for-each.

افتراضياً يتم الفرز بالترتيب الأبجدي للمفاتيح فإذا ظهر أكثر مـن عنصـر xsl:sort فـي عنصر xsl:apply-templates أو عنصر xsl:for-each فإن المخرجات تفرز أولاً بواسـطة أول مفتاح "حرف" ثم ثاني مفتاح "حرف" وهكذا فإذا كان هناك عناصر ما زالت متساوية فـي المقارنة فإنها توضع بالترتيب الذي تظهر به في مستند المصدر.

فعلى سبيل المثال لنفترض وجود ملف ملء بعناصر ATOM مرئية وفقاً للـترتيب الـهجائي وليتم فرزها وفقاً للأعداد الذرية تستخدم ورقة النمط في تعليمات برمجة ١٤-١٨.

تعليمات البرمجة ١٤-١٨: ورقة نمط XSL تفرز بواسطة العدد الذري

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="PERIODIC_TABLE">
    <html>
      <head>
        <title>Atomic Number vs. Atomic Weight</title>
      </head>
      <body>
```

```
      <h1>Atomic Number vs. Atomic Weight</h1>
      <table>
        <th>Element</th>
        <th>Atomic Number</th>
        <th>Atomic Weight</th>
        <xsl:apply-templates>
          <xsl:sort select="ATOMIC_NUMBER"/>
        </xsl:apply-templates>
      </table>
    </body>
  </html>
</xsl:template>

<xsl:template match="ATOM">
  <tr>
    <td><xsl:apply-templates select="NAME"/></td>
    <td><xsl:apply-templates select="ATOMIC_NUMBER"/></td>
    <td><xsl:apply-templates select="ATOMIC_WEIGHT"/></td>
  </tr>
</xsl:template>

</xsl:stylesheet>
```

يوضح الشكل ١٤-٥ النتائج التي توضح حدود الفرز الهجائي ويكون الهيدروجين وله العـدد الذري ١ هو العنصر الأول ومع ذلك فإن العنصر الثاني ليس الهليوم على الرغم من أن عـدده الذري ٢ ولكنه النيون حيث رقمه الذري ١٠ وعلى الرغم من أن رقم ١٠ يكون بده رقـم ٩ إلا أنه أبجدياً الرقم ١٠ يكون قبل الرقم ٢

Atomic Number vs. Atomic Weight - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: file:///D%7C/XML/Bible/14/14-13.html

Atomic Number vs. Atomic Weight

Element	Atomic Number	Atomic Weight
Hydrogen	1	1.00794
Neon	10	20.1797
Fermium	100	257
Mendelevium	101	258
Nobelium	102	259
Lawrencium	103	262
Rutherfordium	104	261
Dubnium	105	262
Seaborgium	106	263
Bohrium	107	262
Hassium	108	265
Meitnerium	109	266
Sodium	11	22.98977
ununnilium	110	269

Document: Done

الشكل ١٤-٥ فرز الذرات هجائياً باستخدام الأعداد الذرية.

يمكن ضبط ترتيب الفرز بواسطة إعداد سمة data-type الاختيارية لقيمة number وعلى سبيل المثال:

```
<xsl:sort data-type="number" select="ATOMIC_NUMBER"/>
```

يوضح الشكل ١٤-٦ العناصر وقد تم فرزها بصورة صحيحة.

يمكن تغيير ترتيب الفرز من الترتيب التصاعدي الافتراضي إلى الترتيب التنازلي عن طريق إعداد سمة order على descending كما يلي:

```
<xsl:sort order="descending"
        sort="number"
        select="ATOMIC_NUMBER"/>
```

تقوم التعليمات السابقة بفرز العناصر من العدد الذري الأكبر إلى الأصغر لهذا يظهر الهيدروجين في أخر القائمة.

Atomic Number vs. Atomic Weight - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: file:///D%7C/XML/Bible/14/14-13.html

Atomic Number vs. Atomic Weight

Element	Atomic Number	Atomic Weight
Hydrogen	1	1.00794
Helium	2	4.0026
Lithium	3	6.941
Beryllium	4	9.01218
Boron	5	10.811
Carbon	6	12.011
Nitrogen	7	14.0067
Oxygen	8	15.9994
Fluorine	9	18.9984
Neon	10	20.1797
Sodium	11	22.98977
Magnesium	12	24.305
Aluminum	13	26.98154
Silicon	14	28.0855

Document: Done

الشكل ١٤-٦ فرز الذرات عددياً باستخدام الأعداد الذرية.

يعتمد الفرز الهجائي على الهجاء ويمكن لسمة lang إعداد لغة المفاتيح ويجب أن تكون قيمة هذه السمة ISO تعليمات برمجة لـ ٦٣٩ مثل en للغة الإنجليزية.



هذه هي نفس القيم المدعمة بواسطة سمة xml:lang التي يناقشها الفصل ١ "مقدمة عن XML".

يمكن إعداد سمة case-order لإحدى القيمتين التاليتين وهما إما upper-first or lower-first لتحديد ما إذا كان الفرز سيكون للأحرف اللاتينية الصغيرة أو العكس ويعتمد الخيار الافتراضي على اللغة نفسها.

علامات CDATA و <

لا تحتوي لغة XSL القياسية على أي وسيلة إدراج أحرف < التي لا تكون جزء من علامة في مستند المخرجات وتجعل علامات أصغر من مستند المخرجات غير صحيح التكوين وهذا ما لا تسمح به لغة XSL وبدلاً من ذلك إذا تم استخدام مرجع حرف مثل < أو مرجع entity مثل < لإدراج الحرف < يقوم المنسق بإدراج < أو <.

تظهر أهمية هذا عند تضمين JavaScript في صفحة وذلك لأن لغة JavaScript تستخدم حرف > لتعني بها رقم أصغر من بدلاً من استخدامها في أية العلامة بالإضافة إلى ذلك لا تتيح تطبيقات JavaScript استبدال حرف > بـ HTML entity وهو &lt.

تدرج علامات < و=< في مستند المخرجات وبالتالي إذا تطلب الأمر استخدام لغة JavaScript في مستند المخرجات وتقوم لغة JavaScript بعمل مقارنات عددية فيمكن كتابة مقارنة أصغر من مقارنات أكبر من أو يساوي عن طريق عكس ترتيب المعامل ويمكن إعادة كتابة مقارنة أصغر من أو يساوي كمقارنة أكبر من وما يلي يوضح بعض الأسطر من تعليمات برمجة للغة JavaScript يستخدمها المؤلف في صفحات الويب.

```
if (location.host.tolowercase().indexof("sunsite"( > 0) {
  location.href="http://metalab.unc.edu/xml/";
}
```

تعتبر تلك الأسطر غير صحيحة التكوين بسبب وجود علامة أصغر من في أول سطرين ومع ذلك فالأسطر السابقة معادلة تماماً للأسطر التالية:

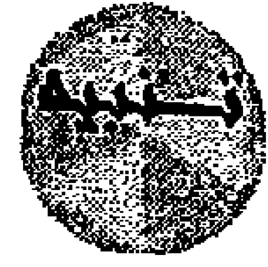
```
if (0 > location.host.tolowercase().indexof("sunsite")) {
  location.href="http://metalab.unc.edu/xml/";
}
```

إذا كان هناك اختبارات متعددة ترتبط بعوامل التشغيل المنطقية قد ترغب في تغيير "و" المنطقية إلى "أو" المنطقية ويوضح المثال التالي أن سطري JavaScript يعتبروا بفاعلية ما إذا كان موقع الصفحة ليس على موقع metalab أو موقع sunsite.

```
if (location.host.toLowerCase().indexOf("metalab"( > 0
 && location.host.tolowercase().indexof("sunsite"( > 0) {
   location.href="http://metalab.unc.edu/xml/";
}
```

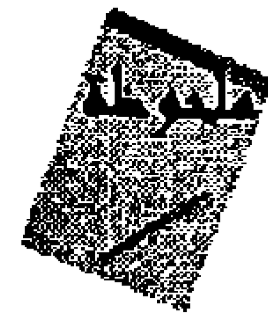
تعتبر تلك السطر غير صحيحة التكوين بسبب وجود علامات أصغر من في أول سطرين ومع ذلك الأسطر التالية التي تختبر ما إذا كانت الصفحة في موقع metalab أو موقع sunsite تعامل الأسطر السابقة تماماً.

```
if (0 > location.host.toLowerCase().indexOf("metalab")
 || 0 > location.host.tolowercase().indexof("sunsite")) {
   location.href="http://metalab.unc.edu/xml/";
}
```



يمكن وضع JavaScript في مستند منفصل ويتم عمل ارتباط إليه من سمة SRC لعنصر SCRIPT ومع ذلك لم يكن هذا إجراء يعتمد عليه قبل ظهور Internet Explorer 4 و Netscape Navigator 3.

لا يسمح بوجود مقاطع CDATA في مستند المخرجات لضمان البساطة ويمكن كذلك استبدال مقاطع CDATA بمجموعة معادلة من بيانات الأحرف Unicode حيث تتجاوز عن المشكلات التي تنتج عن وجود أحرف مثل < و& وتناسب مقاطع CDATA من يقومون بكتابة مستند ملفات XML بأسهم أما بالنسبة لمنتقى برامج أجهزة الكمبيوتر مثل XSL فهم لا يحتاجونها.



يدعم منسق XSL المضمن في Internet Explorer 5.0 عنصر xsl:cdata غير قياسي لإدراج مقاطع CDATA في مستند المخرجات ومع ذلك فمن غير المتوقع إضافة هذا إلى لغة XSL القياسية وقد يتم إلغاءها من الإصدارات اللاحقة لبرنامج Internet Explorer.

سمات الموضع

ترغب في بعض الأحيان في تضمين نفس المحتوى من مستند المصدر في مستند المخرجات عدة مرات، مرة في كل مكان تريد ظهور البيانات فيه ولكن لنفترض أنه يجب تنسيق البيانات بشكل مختلف في مواقع مختلفة فهذا يتطلب بعض المجهود.

فعلى سبيل المثال، لنفترض أن مستند مخرجات معالجة الجدول الدوري تظهر في سلسلة مكونة من ١٠٠ ارتباط إلى تفاصيل أكثر بشأن كل ذرة على حدا وفي تلك الحالة يبدأ مستند المخرجات كما يلي:

```
<UL>
<LI><A HREF="#Ac">Actinium</A></LI>
<LI><A HREF="#Al">Aluminum</A></LI>
<LI><A HREF="#Am">Americium</A></LI>
<LI><A HREF="#Sb">Antimony</A></LI>
<LI><A HREF="#Ar">Argon</A></LI>
...
```

يظهر الوصف الدقيق للذرة فيما بعد في المستند وقد تم تنسيقه كما يلي:

```
<H3><A NAME="Al">Aluminum</A></H3><P>
   Aluminum
```

```
26.98154
13
3
2740
933.5
Al

  2.7

</P>
```

يكون هذا النوع من التطبيق متاح في أي وقت تم فيه إنشاء نص مترابط لجدول المحتويات أو الفهرس بصورة آلية ويجب تنسيق عنصر NAME للذرة بشكل مختلف في جدول المحتويات عن شكلها في المستند نفسه ولذلك يحتاج المطور إلى قاعدتين مختلفتين على أن يتم تطبيق كليهما على عنصر ATOM في أماكن مختلفة في المستند وحل تلك المشكلة يكون إعطاء كل قاعدة منهما سمة mode لعنصر xsl:apply-templates وهذا ما توضحه تعليمات برمجة ١٤-١٩.

تعليمات البرمجة ١٤-١٩: ورقة نمط XSL تستخدم سمات الوضع لتنسيق نفس البيانات بصورة مختلفة في مكانين مختلفين

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

 <xsl:template match="/PERIODIC_TABLE">
  <HTML>
   <HEAD><TITLE>The Elements</TITLE></HEAD>
   <BODY>

    <H2>Table of Contents</H2>
    <UL>
     <xsl:apply-templates select="ATOM" mode="toc"/>
    </UL>

    <H2>The Elements</H2>
```

```
      <xsl:apply-templates select="ATOM" mode="full"/>

     </BODY>
   </HTML>
 </xsl:template>

 <xsl:template match="ATOM" mode="toc">
   <LI><A>
     <xsl:attribute name="HREF">#<xsl:value-of
       select="SYMBOL"/></xsl:attribute>
     <xsl:value-of select="NAME"/>
   </A></LI>
 </xsl:template>

 <xsl:template match="ATOM" mode="full">
   <H3><A>
     <xsl:attribute name="NAME">
       <xsl:value-of select="SYMBOL"/>
     </xsl:attribute>
     <xsl:value-of select="NAME"/>
   </A></H3>
     <P>
       <xsl:value-of select="."/>
     </P>
 </xsl:template>

</xsl:stylesheet>
```

تعريف الثوابت باستخدام xsl:variable

تساعد الثوابت المسماة في اختصار التعليمات البرمجية حيث يمكن أن يستبدلوا النص المتـــداول الشائع الاستخدام باسم بسيط ومرجع وتسهل الثوابت المسماة ضبط النص المتداول الذي يظهر في مواقع متعددة عن طريق تغيير التعريف الثابت.

يعرف عنصر xsl:variable سلسلة مسماة للاستخدام في أي مكان في ورقة النمط من خلال قالب قيمة سمة ويعتبر عنصر xsl:variable عنصر فارغ يظهر كتابع مباشر لعنصر xsl:stylesheet وله سمة واحدة هي name وتقوم تلك السمة ليتم اسم الإشارة إلى المتغير به وتقوم محتويات عنصر xsl:variable النص المستبدل فعلى سبيل المثال يعرف عنصر xsl:variable متغير باسم copy99 وقيمة Copyright 1999 Elliotte Rusty Harold.

```
<xsl:variable name="copy99">
 Copyright 1999 Elliotte Rusty Harold
</xsl:variable>
```

للوصول إلى قيمة هذا المتغير فيجب سبقها بعلامة الدولار $ لاسم المتغير ويجب استخدام قالب قيمة سمة لإدراج ذلك في سمة.

```
<BLOCK COPYRIGHT="{$copy99}">
</BLOCK >
```

تستخدم xsl:value-of لإدراج نص المتغير البديل في مستند المخرجات كنص:

```
<xsl:value-of select="$copy99"/>
```

قد تحتوي محتويات xsl:variable على ترميز بما في ذلك تعليمات XSL أخرى وهذا يعني أنه يمكن حساب قيمة المتغير بناء على معلومات أخرى متضمناً قيمة المتغيرات الأخرى ومع ذلك فقد لا يشير المتغير إلى نفسه بصورة متصلة سواء بطريقة مباشرة أو غير مباشرة فمثلاً المثال التالي به خطأ.

```
<xsl:variable name="GNU">
 <xsl:value-of select="$GNU"/>'s not Unix
</xsl:variable>
```

أيضاً قد لا يشير متغيرين إلى بعضهما البعض في أسلوب دائري مثل هذا:

```
<xsl:variable name="Thing1">
 Thing1 loves <xsl:value-of select="$Thing2"/>
</xsl:variable>

<xsl:variable name="Thing2">
 Thing2 loves <xsl:value-of select="$Thing1"/>
</xsl:variable>
```

القوالب المسماة

تعتبر المتغيرات مقصورة على النص الأساسي والترميز وتقدم لغة XSL وحدة ماكرو يستطيع جمع النص والترميز القياسي أثناء تغيير البيانات ولنفترض أنك تريد تنسيق العدد الذري والوزن الذري وقيم رئيسية أخرى مثل خلية الجدول بخط صغير نوعه سميك Times باللون الأزرق أي تريد ظهور المخرجات بالمظهر التالي:

```
<td>
  <font face="Times, serif" color="blue" size="2">
    <b>52</b>
  </font>
</td>
```

يمكن تضمين كل ذلك في قاعدة قالب مثل ما يلي:

```
<xsl:template match="ATOMIC_NUMBER">
  <td>
    <font face="Times, serif" color="blue" size="2">
      <b>
        <xsl:value-of select="."/>
      </b>
    </font>
  </td>
</xsl:template>
```

يمكن تكرار هذا الترميز كقالب لقواعد جزئي أو كجزء من القالب المستخدم في قواعد أخرى وعندما يصبح الترميز المفصل أكثر تعقيداً وعندما يظهر في أماكن متعددة مختلفة في ورقة النمط قد يرغب المطور في تحويلها إلى قالب مسمى وتتشابه القوالب المسماة مع المتغيرات ومع ذلك فهم يتيحوا تضمين بيانات من مكان تطبيق القاعدة بدلاً من مجرد إدراج نص ثابت.

قد يكون لعنصر xsl:template سمة name حيث يتم وضعها حيز التميز بواسطة تلك السمة حتى وإن كانت لا تطلق بصورة مباشرة ويوضح المثال التالي عينة قالب مسمى للنموذج السابق.

```
<xsl:template name="ATOM_CELL">
  <td>
    <font face="Times, serif" color="blue" size="2">
      <b>
```

```
      <xsl:value-of select="."/>
     </b>
    </font>
   </td>
  </xsl:template>
```

يستبدل <xsl:value-of select="."/> الموجود في وسط وحدة الماكرو بمحتويات نقطـــة البدء الحالية التي تم استدعاء هذا القالب منها.

يظهر عنصر xsl:call-template في محتويات قاعدة القالب وله وسيطه name مطلوبـــة التي تقوم بإطلاق اسم على القالب الذي يتم استدعائه وعند معالجته يســـتبدل عنصـــر xsl:call-template بمحتويات عنصر xsl:template الذي يقوم بتسميته فعلى سبيل المثال، يمكـــن الآن إعادة كتابة قاعدة ATOMIC_NUMBER فهذا الأسلوب باستخدام عنصـــر xsl:call-template لاستدعاء قالب تسمية ATOM_CELL.

```
<xsl:template match="ATOMIC_NUMBER">
 <xsl:call-template name="ATOM_CELL"/>
</xsl:template>
```

يختصر هذا المثال البسيط فقط بعض الأسطر البسيطة في التعليمات البرمجية ولكن كلما كـــان القالب معقداً وكلما تزايدت مرات استخدامه كلما انخفض مدى تعقد ورقة النمط ومـــن ممـــيزات قوالب التسمية مثلها في ذلك مثل المتغيرات أنها توضح النماذج المتشابهة في ورقة النمـــط ليتـــم تحريرهم كلهم في آن واحد فعلى سبيل المثال في حالة الرغبة في تغيير لون العدد الذري والوزن الذري وكذلك أي قيم رئيسية أخرى من الأزرق إلى الأحمر مل ما يجب فعله هو تغييره مرة في قالب التسمية وليس في كل قاعدة قالب وهذا يضمن ثبات النمط على المدى الطويل.

المعاملات

يمكن لكل تنفيذ منفصل لقالب تسمية تمرير معاملات للقالب لتخصيص مخرجاته ويتـــم تشـــغيل المعاملات في عنصر xsl:template كعناصر تابعـــة xsl:param-variable وفـــي عنصـــر xsl:call-template يتم تمثيل المعاملات كعناصر تابعة xsl:param.

لنفترض على سبيل المثال أنه يجب تضمين ارتباط إلى ملف محدد لكـــل خليـــة ذرة وتبـــدو المخرجات كما يلي:

```
<td>
 <font face="Times, serif" color="blue" size="2">
  <b>
```

```
      <a href="atomic_number.html">52</a>
    </b>
  </font>
</td>
```

يجب تمرير قيمة سمة href من نقطة بدء تشغيل القالب لأن تلك السمة تتغير حسب نقطة بدء تشغيلها في القالب وعلى سبيل المثال يجب تنسيق الأوزان الذرية كما يلي:

```
<td>
  <font face="Times, serif" color="blue" size="2">
    <b>
      <a href="atomic_weight.html">4.0026</a>
    </b>
  </font>
</td>
```

يبدو القالب الذي يدعم ذلك مثل ما يلي:

```
<xsl:template name="ATOM_CELL">
  <xsl:param-variable name="file">
   index.html
  </xsl:param-variable>
  <td>
    <font face="Times, serif" color="blue" size="2">
      <b>
        <a href="{$file}"><xsl:value-of select="."/></a>
      </b>
    </font>
  </td>
</xsl:template>
```

تعطي سمة name لعنصر xsl:param-variable المعامل اسم ويعتبر ذلك شيئاً مهماً فـــي حالة وجود وسائط متعددة وتردد محتويات عنصر xsl: param-variable قيمة افتراضية لـــهذا المعامل ليتم استخدامها إذا لم يقدم استدعاء تشغيل القالب قيمة ويمكن إعطاء ذلك كسلسلة تعبـــير باستخدام سمة expr تماماً مثل xsl:variable.

عند استدعاء هذا القالب يقدم تابع xsl:param لعنصر xsl:call-template قيمة المعـــامل باستخدام سمة name لتعريف المعامل ومحتوياته لتقديم قيمة للمعامل والمثال التالي يوضح ذلك:

```
<xsl:template match="ATOMIC_NUMBER">
  <xsl:call-template macro="ATOM_CELL">
    <xsl:param name="file">atomic_number.html</xsl:param>
    <xsl:value-of select="."/>
  </xsl:call-template>
</xsl:template>
```

ويعتبر المثال السابق بسيطاً ولكن يوجد قوالب تسمية أكثر تعقيداً ممثلاً يمكن تعريف رأس وتذييل وحدات الماكرو للصفحات على مواقع الويب ليتم استيرادها باستخدام أوراق نمط متعددة حيث تقوم كل ورقة نمط بتغيير عدة معاملات لاسم ناشر الصفحة وعنوان الصفحة وتاريخ حق الطبع.

إزالة وحفظ المسافات البيضاء

نسقت مخرجات الأمثلة بطريقة مختلفة بعض الشيء عن المعتاد والسبب في ذلك هو تقييم الأسطر الطويلة في مستند المصدر إلى أسطر متعددة لتلائم حجم الكتاب وللأسف فإن المسافات البيضاء الزائدة التي أضيفت إلى مستند الإدخال ظهرت كذلك في مستند المخرجات بالنسبة لجهاز الكمبيوتر لا تعتبر المسافات البيضاء ذات أهمية أما بالنسبة للشخص العادي فقد تسبب له بعض التشتت.

السلوك الافتراضي لنقاط بدء النص مثل محتوى عنصر ATOMIC_NUMBER وDENSITY هو حفظ كل المسافات البيضاء ويبدو عنصر DENSITY المعتاد مثل ما يلي:

```
<DENSITY UNITS="grams/cubic centimeter"><!- At 300K ->
  7.9
</DENSITY>
```

عند تقدير قيمته يتم تضمين المسافات البيضاء في المقدمة وفي النهاية على الرغم من وجود المسافات البيضاء ليمكن وضع التعليمات داخل الكتاب وليس لها استخدام أخر

```
7.9
```

هناك استثناء واحد وهو في حالة احتواء نقطة بدء النص على مسافات بيضاء فقط ولا يوجد نص أخر ففي تلك الحالة تعتبر المسافة غير هامة ويتم حذفها ولكن هناك استثناء للحالة الاستثنائية وهي إذا كان للنص أصل له سمة xml:space وقيمة هي الحفظ ففي تلك الحالة لا يتم حذفها إلا إذا كان أصل أقرب يحتوي على سمة xml: space بالقيمة الافتراضية ومع ذلك فهذا يبدو أسهل مما هو متصور فكل ما يعنيه ما سبق أنه يمكن تجاهل نقاط بدء النص التي تحتوي على مسافات بيضاء فقط إلا إذا كان قد تم إعدادهم ليكون لهم مسافات بيضاء ذات معنى بخلاف ذلك يتم الاحتفاظ بالمسافات البيضاء.

إذا لم يكن أي من العناصر في المستند سيحتفظ بالمسافات البيضاء يمكن إعداد سمة default-space الخاصة بعنصر xsl:stylesheet على strip وتتم إزالة كافة المسافات البيضاء السابقة واللاحقة من نقاط بدء النص قبل وضعهم في مستند المخرجات ويعتبر هذا أسهل طريقة لحل مشكلة الجدول الدوري ومثال على ذلك:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
                default-space="strip">
```

إذا لم ترغب في إزالة المسافات من كل العناصر تستخدم عناصر xsl:strip-space لتعريف العناصر المحددة في مستند الإدخال التي تعتبر مسافاتها البيضاء غير هامة ولا يتم نسخها إلى مستند المخرجات وتعرف سمة element العنصر الذي يجب إزالة مسافاته الزائدة.

مثال لذلك يمكن إضافة تلك القواعد لورقة نمط الجدول الدوري لتجنب المسافات البيضاء الزائدة.

```
<xsl:strip-space element="DENSITY"/>
<xsl:strip-space element="BOILING_POINT"/>
<xsl:strip-space element="MELTING_POINT"/>
```

يعتبر عنصر xsl:preserve-space هو عكس عنصر xsl:strip-space ويسمى سمة element الخاصة به عنصر يجب حفظ مسافاته البيضاء.

```
<xsl:preserve-space element="ATOM"/>
```

لا تعتبر المسافة البيضاء في ورقة النمط نفسها هامة على عكس المسافات البيضاء في مستند إدخال XML ويتم تخفيضها لمسافة واحدة بصورة افتراضية ويمكن تجنب هذا السلوك عن طريق تضمين المسافات البيضاء للأحرف في عنصر xsl:text وعلى سبيل المثال:

```
<xsl:template select="ATOM">
<xsl:text>     This is indented exactly five spaces.</xsl:text>
</xsl:template>
```

يمكن إلحاق سمة مسافة بادئة إلى النتيجة لعنصر الجذر xsl:stylesheet وإذا كانت لتلك السمة قيمة Yes فينتج المعالج إدراج وليس إزالة المسافات البيضاء الزائدة في مستند المخرجات في محاولة لجعلها تبدو أحسن شكلاً وقد يتضمن ذلك المسافات البادئة والخطوط الفاصلة ويوضح المثال التالي ذلك:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
 indent-result="yes">
  <!- usual templates and such go here... ->
</xsl:stylesheet>
```

في حالة إنتاج مستند HTML يجعل تحديد "indent-result="yes المخرجات مقرونة بصورة مناسبة وتعتبر القيمة الافتراضية indent-result هي "لا" لأن تنسيقات المخرجات التي لا تكون HTML قد تعتبر المسافات البيضاء أكثر أهمية.

انتقاء الاختيارات

تتيح لغة XSL عنصريين لتغيير المخرجات بناء على الإدخالات فعنصر xsl:if إما يضع أو لا يضع كمخرج جزء معطى من XML بناء على النماذج الموجودة في الإدخالات ينتقي عنصر xsl:choose واحد من عدة أجزاء XML متاحة بناء على أي النماذج يكون موجود في الإدخال ويمكن تنفيذ معظم ما يمكن أدائه باستخدام عنصر xsl:if وعنصر xsl:choose بواسطة تطبيقات مناسبة من القوالب ولكن الحلول بواسطة عنصر xsl:if وعنصر xsl:choose أسهل بكثير.

عنصر xsl:if

يقدم عنصر xsl:if ميزة بسيطة لتغيير المخرجات اعتماداً على نموذج وتحتوي سمة test لعنصر xsl:if على تغيير محدد يتم تقديمه إلى عمليات منطقية Boolean فإذا كان التغيير صحيح توضع محتويات عنصر xsl:if كمخرجات إما عكس ذلك فلا يتم وضعها كمخرجات يكتب القالب التالي كل عناصر ATOM ATOM وتضاف فاصلة ومسافة بعد كل عناصر ما عدا العنصر الأخير في القائمة.

```
<xsl:template match="ATOM">
  <xsl:value-of select="NAME"/>
  <xsl:if test="not(position()=last())">, </xsl:if>
</xsl:template>
```

يضمن هذا الإجراء ظهور العناصر مثل "Hydrogen, Helium" وليس "Hydrogen, Helium".

لا توجد عناصر xsl:else وxsl:else يقدم عنصر xsl:choose هذه الميزة.

عنصر xsl:choose

يحدد عنصر xsl:choose أحد المخرجات المتاحة بناء على عدة شروط متوفرة ويقدم عنصر تابع xsl:when كل شرط مع قالب المخرجات المرتبط به أما بالنسبة لسمة test لعنصر xsl:when فهي تعبير محدد له قيمة Boolean في حالة صحة أكثر من شرط يتم

instantiated وإذا لم يكن أي من عناصر xsl:when صحيحة تكون محتويات العناصر التابعة xsl:otherwise وعلى سبيل المثال، تغير تلك القاعدة دون المخرجات بناء على ما إذا كان سمة STATE لعنصر ATOM هي SOLID أو LIQUID أو GAS.

```
<xsl:template match="ATOM">
  <xsl:choose>
    <xsl:when test="@STATE='SOLID'">
      <P style="color:black">
        <xsl:value-of select="."/>
      </P>
    </xsl:when>
    <xsl:when test="@STATE='LIQUID'">
      <P style="color:blue">
        <xsl:value-of select="."/>
      </P>
    </xsl:when>
    <xsl:when test="@STATE='GAS'">
      <P style="color:red">
        <xsl:value-of select="."/>
      </P>
    </xsl:when>
    <xsl:other>
      <P style="color:green">
        <xsl:value-of select="."/>
      </P>
    </xsl:other>
  </xsl:choose>
</xsl:template>
```

دمج أوراق نمط متعددة

يستخدم مستند XML واحد عدة مرادفات ترميز مختلفة تم وصفها في عدة DTD متلفة ويمكن استخدام أوراق نمط قياسية مختلفة لتلك المرادفات المختلفة وبالطبع يجب وجود قواعد نمط لمستندات معينة وتتيح عناصر xsl:import وxsl:include دمج أوراق نمط متعددة ليمكن ترتيب إعادة استخدام أوراق النمط لمرادفات وأهداف مختلفة.

الاستيراد باستخدام xsl:import

يعتبر عنصر xsl:import عنصر عالي المستوى حيث تقدم سمة href التابعة له URI الخاص بأوراق النمط المراد استيرادها ويجب أن تظهر عناصر xsl:import قبل أي عناصر عالية المستوى أخرى في عنصر الجذر xsl:stylesheet ومثال لذلك تستورد عناصر xsl:import أوراق نمط genealogy.xsl و standards.xsl.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
  <xsl:import href="genealogy.xsl"/>
  <xsl:import href="standards.xsl"/>
  <!- other child elements follow ->
</xsl:stylesheet>
```

قد تتعارض القواعد في أوراق النمط التي تم استيرادها مع قواعد في ورقة نمط الاستيراد فإذا حدث ذلك يكون للقواعد في ورقة نمط الاستيراد الأسبقية وإذا تعارضت قاعدتين في أوراق نمط استيراد مختلفة فتأخذ الأسبقية أخر قاعدة تم استيرادها هي في المثال السابق standards.xsl.

تعتبر عناصر xsl:apply-imports مختلفة قليلاً عن عناصر xsl:apply-templates التي تستخدم فقط قواعد الاستيراد فإنها لا تستخدم أي قواعد من ورقة نمط الاستيراد وهذا يتيح الوصول إلى قواعد الاستيراد والتي قد يتم التجاوز عنها في أي حالة أخرى عن طريق القواعد في ورقة نمط الاستيراد وبخلاف الاسم فلها بناء جملة مشابه لعنصر xsl:apply-templates واختلاف الأداء الوحيد هو أنها تطابق قواعد القالب في أوراقث نمط الاستيراد.

التضمين باستخدام xsl:include

يعتبر عنصر xsl:include عنصر عالي المستوى حيث يقوم بنسخ ورقة نمط أخرى إلى ورقة النمط الحالية عند نقطة حدوثها ولتكون أكثر تحديداً فإنه ينسخ محتويات عنصر xsl-stylesheet الموجودة في المستندات البعيدة إلى المستند الحالي وتقدم سمة href الخاصة بها URI الخاصة بورقة النمط لتضمينها يقع عنصر xsl:include في أي مكان على أعلى المستويات بعد عنصر xsl:import الأخير.

على عكس القواعد المضمنة بواسطة عناصر xsl:import فإن القواعد المضمنة باستخدام عناصر xsl:include لها نفس الأسبقية في تضمين ورقة النمط التي قد تحصل عليها إذا نسختهم ولصقهم من ورقة نمط إلى أخرى وفيما يخص محرك التنسيق فلا يوجد اختلاف بين قاعدة مضمنة وقاعدة ظاهرة.

تضمين أوراق نمط في مستندات باستخدام xsl:stylesheet

يمكن تضمين ورقة نمط XSL مباشرة في مستند XML مطبق عليه ولكن لا يفضل تنفيذ ذلك في التطبيق العملي ولا تتطلب المستعرضات ومحركات التنسيق تدعيمهم ولفعل ذلك يجب أن يظهر عنصر xsl:stylesheet كتابع لعنصر المستند بدلاً من عنصر الجذر نفسه وبذلك يكون له سمة id وتعطي اسم مميز وتظهر سمة id كقيمة سمة href في تعليمات معالجة xsl:stylesheet وذلك بعد معرف الارتساء # وتوضح تعليمات برمجة ١٤-٢٠ ذلك:

تعليمات البرمجة ١٤-٢٠: ورقة نمط SXL المضمنة في مستند XML

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="#id(mystyle)"?>
<PERIODIC_TABLE>

  <xsl:stylesheet
   xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
   id="mystyle">

   <xsl:template match="/">
     <html>
       <xsl:apply-templates/>
     </html>
   </xsl:template>

   <xsl:template match="PERIODIC_TABLE">
       <xsl:apply-templates/>
   </xsl:template>

   <xsl:template match="ATOM">
     <P>
       <xsl:value-of select="."/>
     </P>
   </xsl:template>
```

```
</xsl:stylesheet>
<ATOM>
  <NAME>Actinium</NAME>
  <ATOMIC_WEIGHT>227</ATOMIC_WEIGHT>
  <ATOMIC_NUMBER>89</ATOMIC_NUMBER>
  <OXIDATION_STATES>3</OXIDATION_STATES>
  <BOILING_POINT UNITS="Kelvin">3470</BOILING_POINT>
  <MELTING_POINT UNITS="kelvin">1324</MELTING_POINT>
  <SYMBOL>Ac</SYMBOL>
  <DENSITY UNITS="grams/cubic centimeter"><!- At 300K ->
    10.07
  </DENSITY>
  <ELECTRONEGATIVITY>1.1</ELECTRONEGATIVITY>
  <ATOMIC_RADIUS UNITS="Angstroms">1.88</ATOMIC_RADIUS>
 </ATOM>

</PERIODIC_TABLE>
```

خلاصة

تعلمنا في هذا الفصل تحولات XSL وبالتحديد:

- تتكون لغة "Extensible Style Language (XSL)" من تطبيقين مفصلين للغة XML لتمويل وتنسيق مستندات XML.
- تطبق تحولات XSL قواعد على شجرة تقرأ من خلال مستند XML لتحويلها إلى شجرة مخرجات تكيف كمستند XML.
- تعتبر قاعدة قالب XSL عنصر xsl:template مع سمة match وتتم مقارنة نقاط البدء في شجرة الإدخال بالنماذج الموجودة في سمات match لعناصر القوالب المختلفة وعند ظهور مطابقة تصبح محتويات القالب مخرج.
- قيمة نقطة البدء هي سلسلة نصية فقط وليس ترميز وتحتوي على محتويات نقطة البدء ويتم حساب ذلك بواسطة عنصر xsl:value-of.
- يمكن معالجة عناصر متعددة بأسلوبين هما عنصر xsl:apply_templates وعنصر xsl:for each.

- قيمة سمة match لعنصر xsl:template هي نمط مطابق يحدد أي نقاط البدء طابقها القالب.
- تعتبر التعبيرات المحددة مجموعة إضافية لسمة المطابقة المستخدمة بواسطة سمة Select لعناصر xsl:apply-templates وxsl:value-of وxsl:for-each وxsl:copy-of وxsl:sort وغيرهم.
- تطبق قاعدتين افتراضيتين قوالب لنقاط بدء العناصر وتأخذ قيمة نقاط بدء النص.
- يمكن لعناصر xsl:element وxsl:attribute وxsl:pi وxsl:comment وxsl:text وضع عناصر وسمات وتعليمات معالجة وتعليقات ونصوص محسوبة من البيانات في مستند الإدخال في مستند المخرجات.
- يعرف عنصر xsl:attribute-set مجموعة شائعة من السمات الممكن تطبيقها إلى عناصر متعددة في قوالب مختلفة باستخدام عنصر xsl:use.
- ينسخ عنصر xsl:copy نقطة الإدخال الحالية إلى المخرجات.
- يدرج عنصر xsl:number الرقم المحدد بواسطة سمة expr إلى مستند المخرجات باستخدام تنسيق رقمي محدد معطى من قبل سمة format.
- يعيد عنصر xsl:sort ترتيب نقاط بدء الإدخال قبل نسخهم إلى مستند المخرجات.
- لا تستطيع لغة XSL وضع مقاطع CDATA وعلامات (<) الضرورية كمخرجات.
- يمكن لعناصر Modes تطبيق قوالب مختلفة على نفس العنصر من مواقع مختلفة في ورقة النمط.
- يعرف عنصر xsl:variable ثوابت الأسماء التي يمكن بها توضيح التعليمات البرمجية.
- تساعد قوالب التسمية وإعادة استخدام التعليمات البرمجية الشائعة للبرنامج.
- تظل المسافات البيضاء موجودة بصورة افتراضية إلا إذا كان في عنصر xsl:strip-space ما يشير إلى عكس ذلك.
- يخرج عنصر ينتج عنصر xsl:if مخرج إذا كانت سمة test الخاصة به صحيحة.
- يخرج عنصر xsl:choose قالب لأول تابع xsl:when وتكون سمة test له صحيحة أو قالب عنصر xsl:default إذا لم يكن لعنصر xsl:when سمة اختيار true.
- تدمج عناصر xsl:import وxsl:include قواعد من أوراق نمط مختلفة.

نتعلم في الفصل القادم مرادفات تنسيق الكائنات وهي الجزء الثاني من لغة XSL ويعتبر تنسيق الكائنات أسلوب فعال في تحديد التخطيط المحدد للصفحة وتستخدم تحولات XSL لتحويل مستند XML إلى مستند XSL لتنسيق الكائنات.



الفصل

كائنات تنسيق XML

في هذا الفصل

يعد النصف الثاني من XML لغة التنسيق ويستخدم تطبيق من تطبيقات XML لوصف كيفية أداء المحتوى عند تقديمه للقارئ وتستخدم ورقة النمط لغة التحويل لتحول مستند XML إلى مستند XML جديد الذي يستخدم معجم كائنات التنسيق بينما هناك أمال كثيرة بأن تعرف كيف يعرض مستعرض الويب البيانات مباشرة. وقد تم ترميزها بكائنات تنسيق XML ولأن هناك خطوة إضافية أساسية والتي فيها مخرج المستند تم تحويلها إلى تنسيق آخر مثل PDF.

نظرة عامة عن لغة تنسيق XSL

تزودك كائنات تنسيق XSL بنموذج تخطيط مرئي أكثر تعقيداً من HTML+CSS وهو التنسيق المدعم بواسطة كائنات التنسيق غير أنه لم يتم تدعيمها بواسطة HTML+CSS بما في ذلك التخطيط غير المرئي والملحوظات وأرقام الصفحات وما إلى ذلك خاصة عندما يكون CSS مهيأ للاستخدام على الويب فإن كائنات تنسيق XSL تم تصميمها لأكثر من استخدام عام يجب كتابة ورقة نمط XSL التي تستخدم كائنات التنسيق لتخطيط كتاب مطبوع بأكمله وورقة النمط المختلفة التي يجب أن يكون لديها القدرة على تحويل مستند XML إلى موقع الويب.

تنبيه حول لغة تنسيق XSL

ما يزال XSL تحت التطوير وقد تغير وسيتغير في المستقبل وقد استند هذا الفصل على مسودة من تحديدات XSL وأثناء قراءتك لهذا الكتاب ستجد أن مسودة XSL قد تم نسخها وأن بناء جملة XSL قد تم تغييرها. كائنات التنسيق جزء من تحديد هو حتى إذا كان لم يكن هناك أي شيء أقل اكتمال من تحديد لغة التحول. إذا واجهتك شيء لا يعمل بالشكل الصحيح ربما أردت مقارنة الأمثلة في هذا الكتاب وأحدث التحديدات.

وربما يزيد الأمر سوءاً هو أنه ليس هناك برامج تنفذ كل مواصفات مسودة التي تم العمل بها في ٢١ إبريل من عام ١٩٩٩ المتعلقة بخصائص XSL حتى تصف كائنات التنسيق. في الواقع ليس هناك مستعرض ويب يستطيع عرض أحد المستندات التي تم كتابتها باستخدام كائنات تنسيق XSL.

وفي الواقع تعد هذه من أحد الخطوات نحو التصور النهائي للمنتج النهائي، وحتى هذه المرحلة فإنه يمكنك العمل مع كائنات تنسيق XSL في حالها الحالية على عيوبها وعدم اكتمالها مع محاولة التعامل مع كل العيوب والنقائص الموجودة بها، أو اختيار إحدى التقنيات مثل CSS حتى اكتمال XSL.

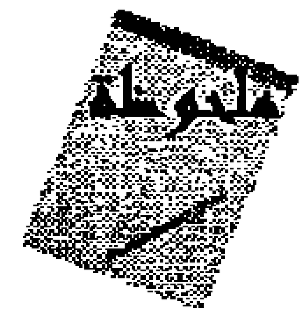
كائنات التنسيق وخصائصها

هناك ٥١ من عناصر كائنات تنسيق XSL أغلبها تحدد أنواع من مساحات مستطيلة والباقي عبارة عن حاويات للمساحات المستطيلة والفراغات. تم وضع كائنات التنسيق في ترتيب أبجدي وهـــي كالتالي:

- Bidi-override
- Block
- Character
- Display-graphic
- display-included-container
- display-rule
- display-sequence
- first-line-marker
- float
- flow
- footnote
- footnote-citation
- inline-graphic
- inline-included-container
- inline-rule
- inline-sequence
- layout-master-set
- list-block
- list-item
- list-item-body
- list-item-label
- multi-case
- multi-properties
- page-number
- page-number-citation
- page-sequence
- region-after
- region-before
- region-body
- region-end
- region-start
- root
- sequence-specification
- sequence-specifier-alternating
- sequence-specifier-repeating
- sequence-specifier-single
- simple-link
- simple-page-master
- static-content
- table
- table-and-caption
- table-body
- table-caption
- table-cell

- multi-property-set
- multi-switch
- multi-toggle
- table-column
- table-footer
- table-header
- table-row

تم عمل تنسيق XSL بناءً على مربعات مستطيلة تسمى مساحات وهي تحوي نص وفـراغ أو كائنات تنسيق، كما هو الحال مع مربعات CSS فكل مساحة لها حد ومساحة على الجوانب وعلى الرغم من أن هوامش CSS يتم استبدالها بمسافات XSL يقوم منسق XSL بقراءة كائنات التنسـيق لتحديد مكان المساحات على الصفحة. تنتج الكثير من كائنات التنسيق مساحات فردية ولكن بسبب فواصل الصفحة والتفاف الصفحة والواصلة وبعض كائنات التنسيق التي تقوم بتشغيل أكثر مـن مساحة واحدة.



يختلف الصندوق الذي يحتوي على مساحة عن الصندوق الـذي يحتـوي علـى حروف مساحة بيضاء حيث أن الأول يشير إلى مساحة فارغة على الصفحـة أو الشاشة مثل الهوامش على الناحية اليمنى أو اليسرى من هذه الصفحة وهي بالطبع تختلف عن مسافة الأحرف بين الكلمات على هذه الصفحة.

تختلف كائنات تنسيق أساساً في محتواها فمثلاً كائن تنسـيق list-item-label هـو مربـع يحتوي على نقطة أو رقم أو أي مؤثر. وكائن تنسيق list-item-body هو المربع الذي يحتـوي على النص، ويحتوي list-item على كائنات تنسيق list-item-label وlist-item formatting معاً.

وقد قسمت كائنات التنسيق إلى أربع أنواع مختلفة من مساحات المستطيل:

١ – محتويات المساحة

٢ – مساحات الكتل

٣ – مساحات الأسطر

٤ – مساحات محولة

يعتبر هذا ترتيب معقد فمحتويات المساحة تحتوي محتويات مساحة أصغر ومساحات الكتـل ومساحات الكتل تحتوي مساحات كتل أخرى ومساحات خطوط ومحتويات وتحتـوي مسـاحات الخطوط على مساحات محولة وتحتوي المساحات المحولة على مساحات محولة أخرى ولمزيـد من التحديد:

- تعد حاوية المساحة هي أعلى مستوى في XSL ويمكن وضعها عند المنسقات المناسبة داخل المساحة التي تحتويها. يمكن أن تحتوي على حاويات ذات مساحة أصغر أو تابع من مساحات الكتل وعرض المسافتين. يمكن اعتبار إحدى صفحات هذا الكتاب حاوية مساحة يحتوي على خمس حاويات مساحة أخرى ورؤوس الصفحة والصفحة وتذييل الصفحة والهوامش اليمنى "في هذا المثال مساحات الهوامش لا تحتوي على شئ". تتضمن كائنات التنسيق التي تنتج حاويات المساحة region-body, region-before, region-after, region-start, and region-end.
- تمثل مساحة كتلة عنصر مستوى الكتلة مثل فقرة أو عنصر قائمة. وبالرغم من أن مساحات الكتل ربما تحتوي على مساحات كتل أخرى إلا أنه يجب عمل خط فاصل قبل البدء وبعد الانتهاء من كل مساحة كتلة وبالتسلسل في المساحة التي تحوي مساحات كتل أخرى يتم إضافة وإزالة ما بداخله، يتحرك وضع مساحة الكتلة كلما كان ذلك مهماً لخلق مساحة. وربما تحتوي على مساحة الكتلة وعرض المسافات وناحية الكتل التي يتم ترتيبها بشكل متسلسل في مساحة الكتل وربما تحتوي مساحة الكتل على صورة رسم واحد. كائنات التنسيق التي تنتج مساحات كتل تتضمن block, display-graphic, display-link, display-rule, and list-block.
- تمثل مساحة السطر بسطر من نص. ويكون كل سطر منفصل في عنصر القائمة هذه هو مساحة سطر. وبإمكان مساحات السطر أن تحتوي على مساحات محولة وفراغات محولة. لا يوجد كائنات تنسيق تشابه مساحات الأسطر وكبديل فإن محرك التنسيق يحسب مساحات بناء على شكل الأسطر داخل مساحات الكتل.
- المساحة المحولة هي أجزاء من سطر مثل حرف واحد ومرجع حاشية سفلية أو معادلة حسابية. يمكن أن تحتوي مساحات أخرى محولة وفراغات محولة كائنات التنسيق التي تنتج مساحات محولة تضمن include character وinline-graphic وinline-link وinline-rule وinline-sequence وpage-number.

The fo Namespace

يتم وضع كائنات تنسيق XSL لعناصر XML في http://www.w3.org/ XSL/Format/1.0 مكان الاسم مع الإعلان المتوقع في ورقة نمط XSL.

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:fo="http://www.w3.org/XSL/Format/1.0"
  result-ns="fo">
```

وغالباً ما تكون البادئة هي fo وبالتالي يمكنك رؤية العناصر التالية على هذا النحو:

- fo:bidi-override
- fo:block
- fo:character
- fo:display-graphic
- fo:display-included-container
- fo:display-rule
- fo:display-sequence
- fo:first-line-marker
- fo:float
- fo:flow
- fo:footnote
- fo:footnote-citation
- fo:inline-graphic
- fo:inline-included-container
- fo:inline-rule
- fo:inline-sequence
- fo:layout-master-set
- fo:list-block
- fo:list-item
- fo:list-item-body
- fo:list-item-label
- fo:multi-case
- fo:multi-properties
- fo:multi-property-set
- fo:multi-switch
- fo:multi-toggle

- fo:page-number
- fo:page-number-citation
- fo:page-sequence
- fo:region-after
- fo:region-before
- fo:region-body
- fo:region-end
- fo:region-start
- fo:root
- fo:sequence-specification
- fo:sequence-specifier-
- alternating
- fo:sequence-specifier-
- repeating
- fo:sequence-specifier-
- single
- fo:simple-link
- fo:simple-page-master
- fo:static-content
- fo:table
- fo:table-and-caption
- fo:table-body
- fo:table-caption
- fo:table-cell
- fo:table-column
- fo:table-footer
- fo:table-header
- fo:table-row

استخدم البادئة fo في هذا الفصل.



وقد تم شرح Namespaces في ١٨ وحتى ذلك الحين كل ما يجب أن تعرفه هو أن أسماء كل عناصر كائنات تنسيق XSL تبدأ بـــ fo.

خصائص التنسيق

تحدد كائنات التنسيق المتنوعة في مستند XSL الترتيب الذي فيه يتم وضع المحتوى في الصفحة أو الصفحات ومع ذلك فإن كل تفاصيل التنسيق تتضمن ولكنها غير محددة إلى حجم الصفحة أو حجم العنصر أو الخط أو اللون أو يكون الكثير منها محدد بخصائص XSL. يتم وضع خصائص التنسيق هذه كسمات على عنصر كائن التنسيق.

يجب أن تكون تفاصيل الكثير من الخصائص معروفة من CSS فالعمل الذي يؤكد أن CSS وXSL يستخدم نفس الأسماء للإشارة إلى نفس الأشياء، فمثلاً خاصية font-family الموجودة في CSS تعني خاصية font-family الموجودة في XSL . وبالرغم من كون بناء الجملة لتعيين القيم للخصائص مختلف في CSS عنه في XSL إلا أن بناء جملة القيم نفسها واحدة في الاثنين. لتحديد ما إذا كان عنصر fo:block قد تم تنسيقه باستخدام Times استخدم قاعدة CSS هذا.

```
fo:block {font-family: New York, Times New Roman, Times, serif}
```

يقوم المناظر XSL بإضافة سمة font-family في علامة البدءfo:block بهذه الطريقة:

```
<fo:block
  font-family="New York, Times New Roman, Times, serif">
```

بالرغم من أن هذا يعد مختلفاً إلا أن اسم النمط "font-family" وقيمة النمط "New York وTimes New Roman وTimes, serif" تعد واحدة. وتم تحديد خاصية font-family كقائمة من أسماء الخطوط يتم فصلها بفاصلات من الاختيار الأول حتى الأخير يستطيع كل من CSS وXSL فهم الكلمة الأساسية serif لتعني خط serif.



تماماً مثل ما حدث في المسودة الرابعة من تحديد XSL الذي تم إسناد هذا الفصل عليه فإن التزامن بين خصائص CSS وXSL لم يتم الانتهاء منه بعد.

وبالطبع فإن كائنات تنسيق XSL تدعم العديد من الخصائص التي ليس لها مناظر في CSS مثل font-size-adjust وcharacter وhyphenation-keep. ويجب تعلم ذلك حتى يتسنى لك التحكم الكامل في XSL. وهي كما يلي:

- auto-restore
- azimuth
- background
- background-attachment
- background-color
- background-image
- background-position
- background-repeat
- border
- border-after-color
- border-after-style
- border-after-width
- border-before-color
- border-before-style
- border-before-width
- border-bottom
- border-bottom-color
- border-bottom-style
- border-bottom-width
- border-collapse
- border-color
- border-end-color
- border-end-style
- border-end-width
- border-left
- border-left-color
- border-left-style
- margin
- margin-bottom
- margin-left
- margin-right
- margin-top
- max-height
- max-width
- may-break-after-row
- may-break-before-row
- min-height
- min-width
- name
- n-columns-repeated
- n-columns-spanned
- n-digits-per-group
- n-rows-spanned
- orphans
- overflow
- padding
- padding-after
- padding-before
- padding-bottom
- padding-end
- padding-left
- padding-right
- padding-start
- padding-top

- border-left-width
- border-right
- border-right-color
- border-right-style
- border-right-width
- border-spacing
- border-start-color
- border-start-style
- border-start-width
- border-style
- border-top
- border-top-color
- border-top-style
- border-top-width
- border-width
- bottom
- break-after
- break-before
- caption-side
- cell-height
- character
- clear
- clip
- color
- column-count
- column-gap
- column-number
- page-break-inside
- page-height
- page-master-blank-even
- page-master-even
- page-master-first
- page-master-last-even
- page-master-last-odd
- page-master-name
- page-master-odd
- page-master-repeating
- page-width
- pause
- pause-after
- pause-before
- pitch
- pitch-range
- play-during
- position
- precedence
- provisional-distance-between-starts
- provisional-label-separation
- reference-orientation
- ref-id
- richness
- right
- row-height

- column-width
- country
- cue
- cue-after
- cue-before
- digit-group-sep
- direction
- elevation
- empty-cells
- end-indent
- ends-row
- extent
- external-destination
- float
- flow-name
- font
- font-family
- font-height-override-after
- font-height-override-before
- font-size
- font-size-adjust
- font-stretch
- font-style
- font-variant
- font-weight
- format
- rule-orientation
- rule-style
- rule-thickness
- scale
- score-spaces
- script
- sequence-src
- show-destination
- size
- space-above-destination-block
- space-above-destination-start
- space-after
- space-before
- space-between-list-rows
- space-end
- space-start
- span
- speak
- speak-header
- speak-numeral
- speak-punctuation
- speech-rate
- start-indent
- starts-row
- state
- stress

- height
- href
- hyphenate
- hyphenation-char
- hyphenation-keep
- hyphenation-ladder-count
- hyphenation-push-char-count
- hyphenation-remain-char-count
- id
- indicate-destination
- inhibit-line-breaks
- initial
- initial-page-number
- internal-destination
- keep-with-next
- keep-with-previous
- language
- last-line-end-indent
- left
- length
- letter-spacing
- letter-value
- line-height
- line-height-shift-adjustment
- line-stacking-strategy
- switch-to
- table-height
- table-layout
- table-omit-middle-footer
- table-omit-middle-header
- table-width
- text-align
- text-align-last
- text-decoration
- text-indent
- text-shadow
- text-transform
- title
- top
- vertical-align
- visibility
- voice-family
- volume
- white-space-treatment
- widows
- width
- word-spacing
- wrap-option
- writing-mode
- z-index

التحويل إلى كائنات التنسيق

تنتمي كائنات تنسيق XSL لمعجم XML وهي تستخدم لترتيب العناصر على الصفحة. المستند الذي يستخدم كائنات تنسيق XSL هو مستند XML جيد التكوين كما أنه يستخدم هذا المعجم وهذا يعني عمل عنصر جذري وعناصر تابع وهكذا. يجب أن تقترب من كل القواعد الخاصة بالتكوين الجيد الخاصة بأي مستند XML . ويحتوي الملف الذي تم استخدام تنسيق XSL على الامتداد fob في ومع ذلك ربما يكون في أخرها XML لأنه ملف XML الذي تم صناعته بشكل جيد.

تعد تعليمات برمجة ١٥-١ مستند بسيط تم ترميزه باستخدام كائنات XSL وجذر المستند هو fo:root ويحتوي على fo:layout-master-set و fo:page-sequence. يحتوي عنصر fo:layout-master-set على عناصر تابع simple-page-master كل من fo:simple-page-master يصف نوع الصفحة التي يتضح عليها المحتوى. هذه هي صفحة بسيطة إلا أن المستندات الأكثر تعقيداً لديها صفحات رئيسية لصفحة للبداية والنهاية وكل منها لديه مجموعة هوامش مختلفة وترقيم وميزات أخرى.

ويتم وضع المحتوى على نسخ من الصفحات الرئيسية باستخدام fo:page-sequence وتحتوي fo:page-sequence contains على fo:sequence الذي يحدد الترتيب الذي تستخدم فيه الصفحات الرئيسية المختلفة، وبالتالي فإنها تحتوي على العنصر التابع fo:block الذي يقوم بوضع المحتوى الأصلي على الصفحات الرئيسية في تتابع محدد. المحتوى هنا هم العنصران التابعان fo:block وكل لديه خاصية font-size من ٢٠نقطة وخاصية family property من serif.

تعليمات البرمجة ١٥-١: مستند بسيط تم استخدام معجم كائن تنسيق XSL

```
<fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">

  <fo:layout-master-set>
    <fo:simple-page-master page-master-name="only">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence>
```

```
<fo:sequence-specification>
  <fo:sequence-specifier-single page-master-name="only"/>
</fo:sequence-specification>

<fo:flow>
  <fo:block font-size="20pt" font-family="serif">
    Hydrogen
  </fo:block>
  <fo:block font-size="20pt" font-family="serif">
    Helium
  </fo:block>
</fo:flow>

</fo:page-sequence>

</fo:root>
```

بالرغم من إمكانية كتابة مستند مثل ذلك الموجود في تعليمات برمجة ١٥-١ يدوياً والذي سيفقد كل ميزات تنسيق المحتوى ذاتياً التي توفرها XML. ستكتب ورقة نمط XSL التي تستخدم معجم تحويل XSL لتحويل مستند المصدر إلى معجم كائن تنسيق. تعليمات برمجة ١٥-٢ هي ورقة نمط XSL التي قامت بإنتاج تعليمات برمجة ١٥-١ عن طريق تحويل تعليمات برمجة١٤-١ الموجودة في الفصل السابق.

تعليمات البرمجة ١٥-٢: تحويل من معجم مصدر إلى كائنات تنسيق XSL

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
 xmlns:fo="http://www.w3.org/XSL/Format/1.0"
 result-ns="fo" indent-result="yes">

 <xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">
```

```
    <fo:layout-master-set>
      <fo:simple-page-master page-master-name="only">
        <fo:region-body/>
      </fo:simple-page-master>
    </fo:layout-master-set>

    <fo:page-sequence>

      <fo:sequence-specification>
        <fo:sequence-specifier-single
            page-master-name="only"/>
      </fo:sequence-specification>

      <fo:flow>

       <xsl:apply-templates select="//ATOM"/>
      </fo:flow>

    </fo:page-sequence>

  </fo:root>
 </xsl:template>

 <xsl:template match="ATOM">
  <fo:block font-size="20pt" font-family="serif">
    <xsl:value-of select="NAME"/>
  </fo:block>
 </xsl:template>

</xsl:stylesheet>
```

استخدام FOP

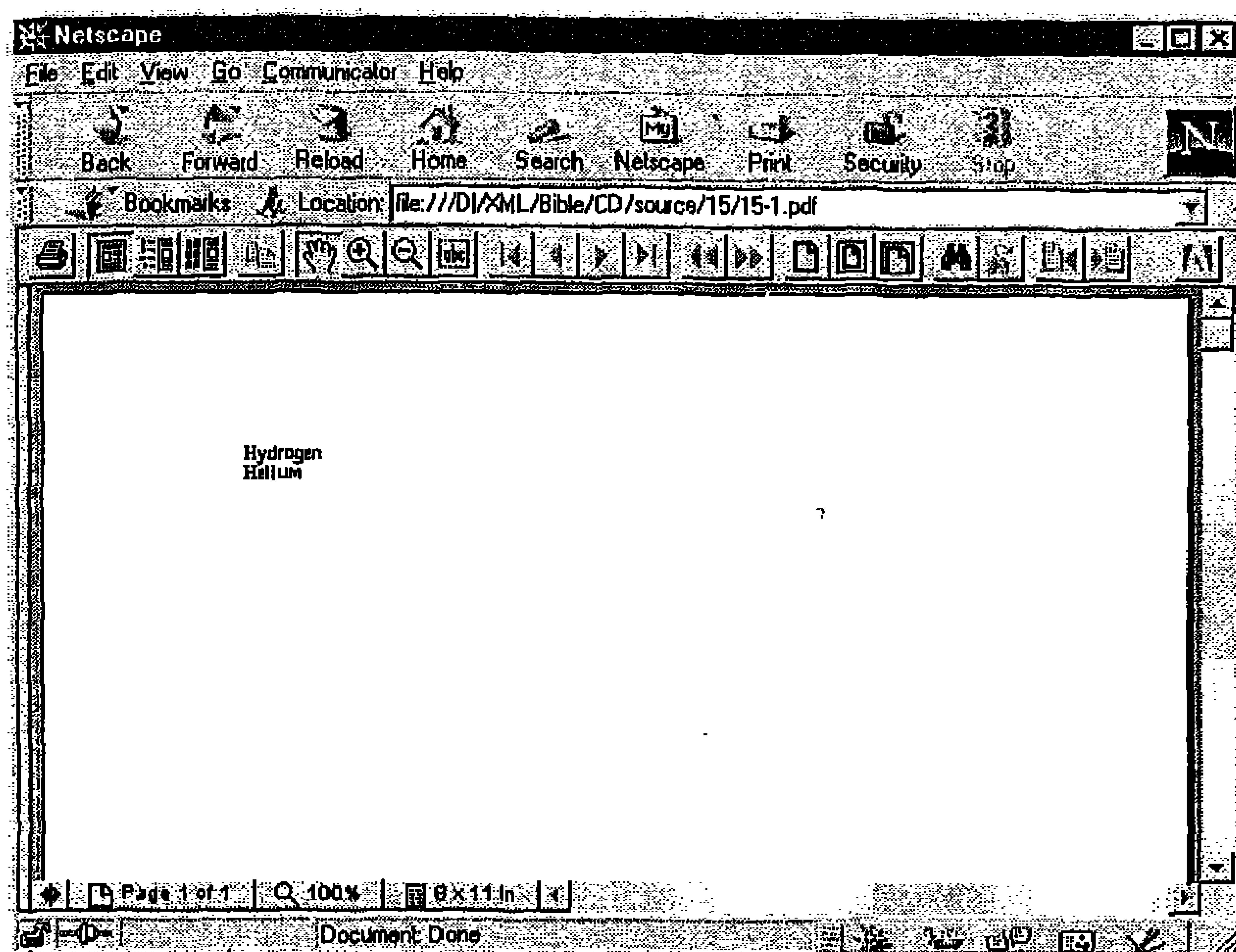
لا يوجد مستعرض يمكنه عرض مستندات XML مباشرة المحولة إلى كائنات XSL حتى الوقت الذي تمت فيه كتابة هذه السطور، ولكن هناك برنامج وحيد يمكنه العمل مع ملف تم ترميزه بكائنات تنسيق XSL وهو FOP. يعد FOP برنامج Java الذي يحول مستندات كائنات التنسيق إلى ملفات Adobe Acrobat PDFK، يمكن تحميل الإصدار الأخر من FOP في http://www.jtauber.com/fop/.

وحتى كتابة هذه السطور فإن الإصدار المتاح من FOP هو 0.6.0 الذي لا يدعم مجموعة فرعية من كائنات التنسيق والخصائص في المسودة الرابعة من XSL. يعد FOP برنامج Java الذي يقوم بتشغيل أي نظام أساسي Java متوافق. عند الرغبة في إعداده قم بوضع أرشيف fop.jar في CLASSPATH الخاص بك الذي يحتوي على فئة com.jtauber.fop.FOP على طريقة ()main. هذا البرنامج قم بتشغيله من سطر أوامر مع تحديد وسيطة ملفات المدخلات والمخرجات فمثلاً:

```
C:\XML\BIBLE\15>java com.jtauber.fop.FOP 15-1.fob 15-1.pdf
James Tauber's FOP 0.6.0
auto page-height: using 11in
auto page-width: using 8in
successfully read and parsed 15-1.fob
laying out page 1...
done page 1.
successfully wrote 15-1.pdf
```

يعد الشكل ١٥-١ fob هو ملف المدخلات الذي يستخدم معجم كائن التنسيق وهي مخرجات ملف PDF الذي يتم عرضه ووضعه بواسطة Adobe Acrobat أو أي برنامج يقوم بقراءة ملفات PDF.

على الرغم من أن ملفات FOP هي نفسها نص ASCII إلا أنه ليست هناك ضرورة للتعرض إلى Post Script لذا من غير المفيد عرض المخرجات الفعلية للمثال السابق ولكن إذا كنت مهتم، افتح ملف FOP في أي محرر نص فيعرض الشكل هذا الملف الذي تم استخراجه وعرضه في Netscape Navigator باستخدام الوظيفة Acrobat.



الشكل ١٥-١ ملف PDF الذي بعد عرضه في Netscape Navigator.

ملفات PDF ليست الوحيدة أو الأولى لمستندات XML التي تم وضعها مع كائنات تنسيق XSL ونأمل في وجود مستعرض يقوم بدعم كائنات تنسيق XSL مستحيلة الترتيب أمـــا الآن فملفـات PDF هي للتنسيق المتاح الوحيد وهذا هو الذي أقدمه في هذا الفصل وبالتالي يجب أن يكون هناك برنامج يمكنه قراءة وعرض هذه الملفات.

تخطيط الصفحة

العنصر الجذري لملف كائنات تنسيق هو fo:root ويحتوي هذا العنصـــر عنصـــر fo:layout-master-set وصفر أو عناصر fo:page-sequence لدى fo:root سمة xmlns:fo ذو قيمة http://www.w3.org/XSL/Format/1.0 وربما لديه سمة id عنصر fo:root موجود فقــط لتحديد namespace وأن يكون المستند الجذري وليس له تأثير مباشر على تخطيط الصفحـــة أو التنسيق.

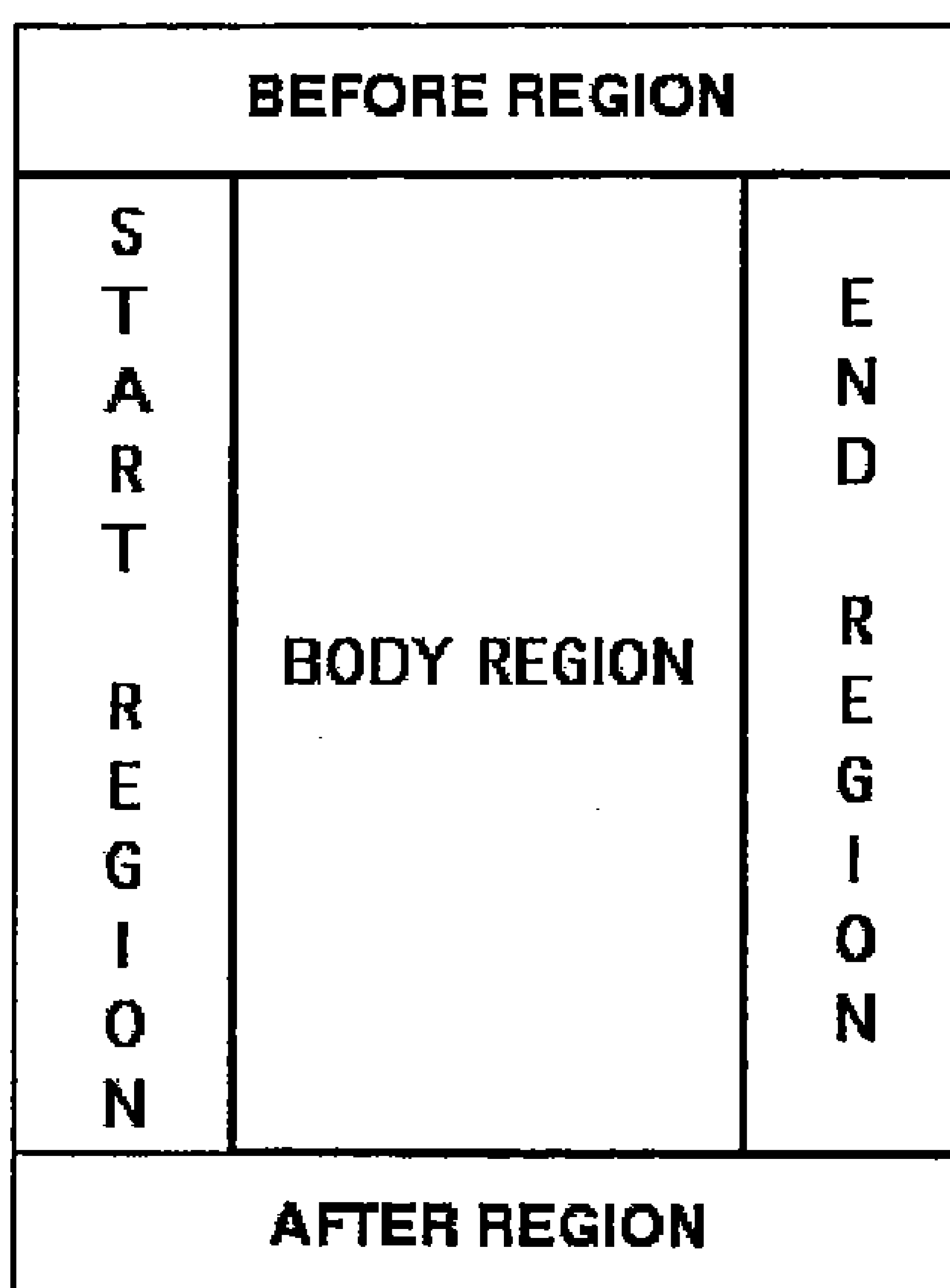
الصفحات الأساسية

عنصر fo:layout-master-set هو لكل الصفحات الرئيسية المختلفـــة المســـتخدمة بواســطة المستند الصفحات الرئيسية البسيطة تشبه أساسها إلى صفحـــات Xpress الرئيســـية أو عـــرض الشرائح باور بوينت كل يحدد التخطيط العام لصفحة متضمنة هوامش وحجم رؤوس الصفحـــات

وتذييل الصفحات ومساحة الصفحة وهكذا وكل صفحة في المستند المستخرج هو مستند على الصفحة على إحدى الصفحات الرئيسية ويضيف بعض الخصائص مثل الهوامش وترقيم الصفحة وتخطيط من الصفحة الرئيسية.

الصفحات الرئيسية البسيطة

كل صفحة رئيسية تمثل بواسطة عنصر fo:simple-page-master ربما تحتوي fo:layout-master-set may إحداهم أو أكثر من هؤلاء يحدد عنصر fo:simple-page-master تخطيط صفحة بما في ذلك حجم قبل المنطقة والمنطقة النهائية والمنطقة الأولية ويعرض الشكل ١٥-٢ التخطيط الأمثل لهذه الأجزاء. الجسم هو كل ما في المنتصف المتروك.



الشكل ١٥-٢ تخطيط أجزاء صفحة واحدة من النص الإنجليزية.



في النص الإنجليزي المنطقة النهائية هي الناحية اليمنى من الصفحة والمنطقة البادئة هي الناحية اليسرى من الصفحة غير أن هذا الوضع يكون معكوساً في النص العبري والعربي ولأن هذه لغات تكتب من اليمين إلى اليسار في أغلب اللغات الحديثة تعد المنطقة السابقة هي رأس الصفحة والمنطقة التالية هي تذييل الصفحات ولكن هذا يمكنه أن يعكس في اللغة التي تكتب من أسفل إلى أعلى.

يقوم المصمم بإعداد حجم منطقة الوسط والرأس والتذييل والمنطقة النهائية والأولى والمسافات بينها باستخدام عناصر حقل المنطقة المناسبة وهي كالتالي:

- fo:region-before
- fo:region-after
- fo:region-body
- fo:region-start
- fo:region-end

ربما تملأ كل من المناطق الخمسة من الصفحة الرئيسية بمحتـــوى مـــن عنصـــر fo:flow وfo:static.

لدى عنصر simple-page-master ثلاث سمات رئيسية:

١ – Page-Master-Name: هو اسم هذه الصفحة الرئيسية التي ستستخدمها تتابع الصفحة لتحديد الصفحة الرئيسية التي ستستخدم على صفحة معينة.

٢ – Page-Height: هو ارتفاع الصفحة.

٣ – Page-Width: هو اتساع الصفحة.

يمكن تجميع Page-Height وPage-Width فرعياً في اختزال منفرد خاصية size وإذا لـم يتم توفير ذلك فإن المنسق يختار افتراض معقول مستند على وسائط المستخدمة. "e.g. 8.5" by "11".

فمثلاً fo:layout-master-set هذا يحتـــوي علـــى عنـــاصر fo:simple-page-master وfo:layout-master-set أحدها للصفحات الزوجية "اليسار" وأحدها للصفحات الفردية وكـــلا منهما لتحديد حجم صفحة يبلغ 8.5 إلى 11 بوصة لبنية هوامش عليا وسفلى مـــن 0.5 بوصـــة وبنية الهامش الداخلي من 0.5 بوصة وهامش خارجي من بوصة واحدة هو شـــائع للصفحـــات المتقابلة.

```
<fo:layout-master-set>
  <fo:simple-page-master page-master-name="even"
    height="8.5in"         width=199in"
    margin-top="5.0in"     margin-bottom="5.0in"
    margin-left="1.0in"    margin-right="5.0in">
   <fo:region-body/>
  </fo:simple-page-master>
  <fo:simple-page-master page-master-name="odd"
    height="8.5in"         width="11in"
```

```
    margin-top="5.0in"    margin-bottom="5.0in"
    margin-left="5.0in"   margin-right="1.0in">
   <fo:region-body/>
  </fo:simple-page-master>
</fo:layout-master-set>
```

تتضمن السمات الأخرى المضافة إلى الصفحات الرئيسية:

- السمات التي تؤثر على هوامش الصفحة margin-bottom, margin-left, margin-right, margin-top, margin
- السمات التي تؤثر على اتجاه الكتابة على الصفحة writing on the page: writing-mode, reference-orientation

خصائص المنطقة

تشترك المناطق الخمسة في نفس الخصائص الأساسية وهي تتضمن:

- السمات التي تحدد كيف يتم تناول المحتوى الذي تجاوز سعته حدود المنطقة: clip وoverflow.
- السمة التي تحدد كيف يتم التفاف المحتوى في الأعمدة: column-count وهي عدد الأعمدة في المنطقة وcolumn-gap وهي المسافة بين الأعمدة.
- السمات التي تؤثر على خلفية المنطقة background, background-attachment, background-color, background-image, background-repeat, background-position.
- السمات التي تؤثر على حدود المنطقة border-before-color, border-before-style, border-before-width, border-after-color, border-after-style, border-after-width, border-start-color, border-start-style, border-start-width, border-end-color, border-end-style, border-end-width, border-top-color, border-top-style, border-top-width, border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border, border-top, border-bottom, border-left, border-right, border-color, border-style, border-width.
- السمات التي تؤثر على المساحة ما بين المحتوى والحدود الخارجية للمنطقة padding-bottom, padding-left, padding-right, padding-top, padding-bottom,

.padding-start, padding-end, padding-before, padding-after, padding

- السمات التي تؤثر على هوامش المنطقة margin-bottom, margin-left, margin-right, margin-top, margin, margin, space-before, space-after, start-indent, end-indent.
- السمات التي تؤثر على اتجاه كتابة المنطقـــة writing in the region: writing-mode, reference-orientation.

أغلب هذه الخصائص يجب أن تكون شبيهة بخصائص CSS من نفس الاســم. ويتــم انتقــاء الخيارات الافتراضية المناسبة لاستخدامها لكل هذه القيم إذا لم يتم إعدادها مسبقاً، وعن طريـــق تثبيتها يمكنك التأثير على التخطيط الكلي للصفحة.

بالإضافة إلى أن المناطق الأربع الخارجية لديها خاصية extent التي تحدد حجــم المنطقــة يحدد حجم الجسم إلا أن حجم المنطقة يبقى كما هو بعد أن يتم حساب المناطق الأربع الأخرى.

فمثلاً هناك fo:layout الذي يجعل كل المناطق الخارجية بوصة واحدة وكـــل منطقــة يتــم إعطاءها حد أسود بمقدار ٢ بكسل والأكثر من ذلك فإن الصفحة نفسها لديها هامش 0.5 بوصـــة على الجوانب.

```
<fo:layout-master-set>
  <fo:simple-page-master page-master-name="only"
     height="8.5in" width="11in"
     margin-top="5.0in"    margin-bottom="5.0in"
     margin-left="1.0in"   margin-right="5.0in">
   <fo:region-start extent="1.0in"
    border-color="black" border-width="2px"/>
   <fo:region-before extent="1.0in"
    border-color="black" border-width="2px"/>
   <fo:region-body
    border-color="black" border-width="2px"/>
   <fo:region-end  extent="1.0in"
    border-color="black" border-width="2px"/>
   <fo:region-after  extent="1.0in"
    border-color="black" border-width="2px"/>
  </fo:simple-page-master>
</fo:layout-master-set>
```

يتم عمل صفحات الجسم بناءً على الصفحة الرئيسية باتساع ٥.٥ بوصة وارتفاع ٨ بوصـــة ويتم حسابها من خلال طرح حجم أي شيء آخر من حجم الصفحة.

تتالي الصفحة

بالإضافة إلى fo:layout-master-set سيحتوي على أحد عناصر fo:layout-master-set أو أكثر فإن كل تتابع صفحة التي تحتوي على شاشات أشياء في الترتيب التالي:

- أحد العناصر fo:sequence-specification الذي تستخدم صفحات الرئيسية.
- عناصر fo:static-content elements التي تحتوي على نص لوضعه في كل صفحة.
- عنصر fo:flow المحتوى على بيانات توضع في كل صفحة تبعاً لدورها.

الاختلاف الرئيسي بين fo:flow وfo:static هو ذاك النص من السريان لم يوضع على أكثر من صفحة حيث المحتوى الثابت، فمثلاً للسطور التي تقرأها الآن هي سريان المحتوى الذي يظهر فقط على هذه الصفحة حيث جزء وعناوين الفصل أعلى الصفحة هي محتوى ثابت يتم تكراره من صفحة إلى صفحة.

يزود fo:sequence-specification بقائمة الصفحات الرئيسية لصف التتابع كل صفحة في التتابع لديها رأس صفحة مرتبط الذي يحدد كيف تبدو الصفحة يستخدم تعليمات برمجة ١٥-١ فقط ليس من الشائع أن يكون لديك المزيد فمثلاً واحد للصفحة الأولى من فصل، أحد الصفحات التالية وأحد كل الصفحات المتتالية مثلاً وما يكون هناك في هذه الحالة فهناك تتابع صفحة كل لجدول المحتوى ونص الجسم المفهرس.

يحتوي عنصر fo:flow وبالترتيب على العناصر الموضوعة على الصفحة كلما تم التعبئة بعناصر من السياق يتم إنشاء الصفحة الجديدة بالتخطيط الرئيسي التالي في تحديد التتابع للعناصر التي لا تزال في السياق.

يحتوي عنصر fo:static-content على معلومات توضح على كل صفحة فمثلاً ربما تضع عنوان الكتاب في رأس كل صفحة يكن ضبط المحتوى على الصفحة الرئيسية فمثلاً جزء من عنصر عنوان fo:static يمكن استخدام عناصر مثل أرقام الصفحات التي تقوم بحسابها من صفحة إلى صفحة عندما تعيد نفس مسألة حسابية في عبارة أخرى ما هو ثابت لا يعد النص ولكن المسألة الحسابية التي تشبع النص.

تحديدات التتالي

يسرد عنصر fo:sequence-specification ترتيب الصفحات الرئيسية ستبدأ باستخدام واحد أو أكثر من هذه العناصر:

fo:sequence-specifier-single
fo:sequence-specifier-alternating

fo:sequence-specifier-repeating

كل هذه العناصر لها سمات التي تحدد أي الصفحات الرئيسية والوقت الذي تستخدم فيه أبسطها هو fo:sequence-specifier-single الذي تحدد سمة page-master-name الصفحة المرئية التي ستبدأ بها فمثلاً تقول هذا العنصر fo:sequence-specification إن المحتوى يجب أن يتم وضعه على مثال واحد من الحرف المسمى في الصفحة الرئيسية.

```
<fo:sequence-specification>
  <fo:sequence-specifier-single page-master-name="letter"/>
</fo:sequence-specification>
```

إذا كان هناك محتوى أكبر من أن تحتويه صفحة مفردة، إذن يتم إما قطع أو تمرير المحتوي الإضافي، بناء على قيم سمات clip وoverflow للمناطق المتعددة حيث يتم وضع المحتوى. مع ذلك، لن يتم إنشاء أكثر من صفحة واحدة. الآن لنرى مواصفات التسلسل التالي:

```
<fo:sequence-specification>
  <fo:sequence-specifier-single page-master-name="letter"/>
  <fo:sequence-specifier-single page-master-name="letter"/>
</fo:sequence-specification>
```

يتاح هذا لصفحات، كل منها بناء على حرف الصفحة الأساسي. إذا تم تعبئة الصفحة الأولى فإنه يتم إنشاء صفحة ثانية وإذا تم تعبئة هذه الصفحة يتم قطع أو التفاف المحتوي.

يمكن استخدام نفس التقنية لتطبيق صفحات أساسية مختلفة، على سبيل المثال، تؤسس مواصفات التسلسل التالية أول صفحة على الصفحة الأساسية المسماة الحرف ١ والصفحة الثانية على الصفحة الأساسية المسماة ٢:

```
<fo:sequence-specification>
  <fo:sequence-specifier-single page-master-name="letter1"/>
  <fo:sequence-specifier-single page-master-name="letter2"/>
</fo:sequence-specification>
```

بالطبع في أغلب الأحيان لن تعرف مسبقاً كم سيكون عدد الصفحات ولذلك تتيح عناصر fo:sequence-specifier-alternating و fo:sequence-specifier-repeating للمستخدم تحديد عدد الصفحات اللازمة لتضمين المحتوى. يحدد عنصر fo:sequence-specifier صفحة أساسية واحدة للصفحة الأولى وصفحة أساسية ثانية لكل الصفحات التالية. يحدد عنصر fo:sequence-specifier-alternating ما يصل إلى ست صفحات أساسية مختلفة للصفحة الأولى وحتى للصفحات الموجود بها محتوى، والصفحات الفردية التي بها محتوى، والصفحات الزوجية الفارغة والصفحات الزوجية الأخيرة والصفحات الفردية الأخيرة.

على سبيل المثال، يحدد التتابع في المثال أن مخرجات الصفحة الأولـــى يجـــب أن تســـتخدم الحرف المسمي للصفحة الأساسية أولا، ولكن كل الصفحات التالية يجب أن تســـتخدم الحـــرف المسمي للصفحة الأساسية.

```
<fo:sequence-specification>
  <fo:sequence-specifier-repeating
    page-master-first="letter_first"
    page-master-repeating="letter"
  />
</fo:sequence-specification>
```

إذا زاد المحتوى الكلي عن الصفحة الأولى سيتم وضعه على الصفحة الثانيـــة، إذا تجـــاوز الصفحة الثانية فإن الصفحة الثالثة سيتم إنشاءها. سيتم تكوين عدد الصفحات المطلـــوب لتحمـــل المحتوى بأكمله.



حتى كتابة هذه السطور لم يحدد بعد ما إذا كان مطلـــوب page-master-first وpage-master-repeating أم لا ومع ذلك إذا كان لديك صفحة رئيسية واحدة ويمكنك بالطبع إعادة استخدامها كقيمة لكل مـــن page-master-first وpage-master-repeating مثل ما يلي:

```
<fo:sequence-specification>
  <fo:sequence-specifier-repeating
    page-master-first="letter"
    page-master-repeating="letter"
  />
</fo:sequence-specification>
```

وقد تم تصميم fo:sequence-specifier-alternating لفصل من كتاب مطبوع حيث تكون لدي الصفحات الأولي والأخيرة بالإضافة إلى الصفحات الفرديـــة والزوجيـــة هوامـــش ورؤوس وتذييل صفحات مختلفة. يكون لدى هذا العنصر سمات تسمح لك بتحديد صفحات رئيســـية لكـــل هذه الصفحات المختلفة، على سبيل المثال:

```
<fo:sequence-specification>
  <fo:sequence-specifier-repeating
    page-master-first="chapter_first"
    page-master-even="chapter_even"
    page-master-blank-even="chapter_blank"
```

```
    page-master-odd="chapter_odd"
    page-master-last-even="chapter_last_even"
    page-master-last-odd="chapter_last_odd"
    page-master-repeating="letter"
  />
</fo:sequence-specification>
```



إذا ظهرت السمات السابقة على أنها ليست متماثلة،على سبيل المثـــال، لا توجـــد سمة page-master-blank-odd، وذلك لأن النشر المعتاد هو غير متمـــاثل. إذا نظرت جيداً إلى صفحات هذا الكتاب أو في أي كتاب أحر ستلاحظ أن الصفحـــات ذات الأرقام الفردية توجد دائماً على اليمين والصفحات ذات الأرقام الزوجية على اليسار وأن الفصول تبدأ دائما على الصفحة اليمنى. يمكن أن تنتهي الفصول إمـــا على الناحية اليمنى "الأرقام الفردية" أو الناحية على الناحيـــة اليســـرى "الأرقـــام الزوجية"، ولكن إذا انتهت على الصفحة الفردية فيتم إدراج صفحة زوجية فارغـــة ليبدأ الفصل التالي على صفحة فردية.

التدفق

يحتوي كائن fo:flow على المحتوى الأصلي الذي يتم وضعه على أمثلة الصفحـــات الرئيســـية المحددة بواسطة مواصفات التتالي. يتكون هذا المحتوى من تتالي مـــن fo:block و-fo:display-graphic وfo:display-link وfo:display-rule وعناصر مستوى كتلة أخرى. في هذا المقطع سنلتزم بعناصر fo:block الأساسية وهي بالتقريب مساوية لعناصر HTML DIV. نرى فيما بعد في هذا الفصل عناصر مستوى كتلة أكثر يمكن أن يحتويها التدفق.

على سبيل المثال، مل يلي هو تدفق أساسي يحتوى على أسماء عدة ذرات، كل فـــي كتلتـــه الخاصة:

```
<fo:flow name="xsl-body">
  <fo:block>Actinium</fo:block>
  <fo:block>Aluminum</fo:block>
  <fo:block>Americium</fo:block>
</fo:flow>
```

تحدد سمة name التي تخص fo:flow وهي في هذا المثال مع قيمـــة xsl-body أي مـــن الخمس مناطق في الصفحة سيتم وضع محتوى التدفق فيها. القيم المسموح بها هي:

- xsl-body

- xsl-after
- xsl-before
- xsl-start
- xsl-end

على سبيل المثال، يكون للتدفق flow بالنسبة لرأس الصفحة "من اليسار إلى اليمين، الأعلى إلى الأسفل في النص الإنجليزي" قيمة flow-name في xsl-before. ما يلي هو تدفق لتذييل الصفحة:

```
<fo:flow id="q2" flow-name="xsl-after">
  <fo:block>
    The XML Bible
    Chapter 15: XSL Formatting Objects
  </fo:block>
</fo:flow>
```

محتوى ثابت

حيث أن كل قطعة من المحتوى من عنصر fo:flow تظهر على صفحة واحدة فإن كل قطعة من محتوى عنصر fo:static-content تظهر على كل صفحة، على سبيل المثال رأس الصفحة أو تذييلها. لا يجب عليك استخدام عناصر fo:static-content ولكن إذا تم استخدامهم يجب أن يظهروا قبل عناصر fo:flow في تتالي الصفحة.

لدى عناصر fo:static-content نفس السمات والمحتويات مثل fo:flow. مع ذلك، لأن fo:static-content لا يستطيع تقسيم محتوياته عبر صفحات متعددة، إذا كان ضروريا، فإنه سيكون لديه محتوى أقل من fo:flow. على سبيل المثال، ما يلي هو fo:static-content لرأس الصفحة:

```
<fo:static-content id="sc2" flow-name="xsl-before">
  <fo:block>
    The XML Bible
    Chapter 15: XSL Formatting Objects
  </fo:block>
</fo:static-content>
```

ترقيم الصفحة

بجانب سمة id المعتادة التي بإمكان أي عنصر كائن تنسيق أن يحصل عليها، فإن عنصر fo:page-sequence لديه ست سمات اختيارية تحدد ترقيم الصفحات للتتالي، وهي:

- initial-page-number
- format
- letter-value
- digit-group-sep
- n-digits-per-group
- sequence-src

تعرف سمة initial-page-number رقم الصفحة الأولى في هذا التتالي. القيمة المحتملة لهذه السمة هي (١)، ولكن قد تكون رقم أكبر إذا كان الصفحات السابقة في ملف مختلف. السمات الخمس المتبقية لديها نفس بناء الجملة والمعنى كما في حالة استخدامها كسمات لعنصر xsl:number من لغة تحويل XSL.



تم مناقشة عنصر xsl:number وسمات format, letter-value, digit-group-sep, n-digits-per-group, sequence-src في مقطع "الأرقام إلى تحويل السلسلة" الفصل ١٤ "لغة تحويل XSL".

كائن تنسيق fo:page-number هو عنصر تحويل فارغ يقوم بأدراج رقم الصفحة الحالية. يكون المنسق مسئول عن تحديد ماهية هذا الرقم. يكون لدى هذا العنصر سمة واحدة فقط هي id. غير ذلك، يمكن تجميع fo:page-number في fo:inline-sequence أوfo:block أو عنصر مشابه لتطبيق خصائص الخط وما إلى ذلك. على سبيل المثال، يستخدم هذا التذييل fo:static-content وfo:page لوضع رقم الصفحة أسفل كل صفحة:

```
<fo:static-content id="sc2" flow-name="xsl-after">
  <fo:block>
    <fo:page-number/>
  </fo:block>
</fo:static-content>
```

يحدد تتالي هذه الصفحة أن رقم الصفحة يستخدم أرقام رومانية صغيرة ويبدأ العد من الرقم ١٠.

```
<fo:page-sequence initial-page-number="10" format="I">

  <!- sequence specification ->

  <fo:static-content flow-name="xsl-after">
```

```
  <fo:block text-align-last="centered" font-size="10 pt">
    <fo:page-number/>
  </fo:block>
 </fo:static-content>

 <!- flows ->

</fo:page-sequence>
```

المحتوى

يكون محتوى "على عكس الترميز" مستند كائنات تنسيق XLS في الأغلب نص. بالإضافة إلى ذلك، يمكن ربط الصور الخارجية إليه بشكل مماثل لعنصر IMG في HTML. يتم حفظ هذا المحتوى في عدة أنواع من العناصر بما في ذلك:

- كائنات تنسيق مستوى الكتلة
- كائنات تنسيق محولة
- كائنات تنسيق جدول
- كائنات تنسيق خارجة عن الخط

تندرج كل هذه الأنواع المختلفة من العناصر إما من عنصر fo:flow أو من عنصر fo:static-content. لا يتم مطلقاً وضعهم مباشرة على الصفحات الأساسية أو تتالي الصفحة.

كائنات تنسيق مستوى الكتلة

يتم رسم كائن تنسيق مستوى الكتلة كمساحة مستطيلة يتم فصلها بخط فاصل وغالبا مساحة بيضاء إضافية من أي محتوى يسبقه أو يتبعه. قد تحتوي الكتل على كتل أخرى، في هذه الحالة يتم أيضا فصل الكتل المحتواة بواسطة خط فاصل وربما مساحة بيضاء إضافية. تتضمن كائنات تنسيق مستوي الكتلة:

- fo:block
- fo:display-graphic
- fo:display-rule
- fo:display-included-container
- fo:display-sequence

- fo:list
- fo:list-item

يعتبر عنصر fo:block هو نظير XSL في display: block في CSS أو DIV فـــي لغــة HTML. قد يتم تضمين الكتل في عناصر fo:flow وعنـــاصر fo:block أخــرى، وعنـــاصر fo:static-content. قد تحتوي عناصر fo:block على عناصر fo:block أخرى، وعنـــاصر مستوى كتلة أخرى مثل fo: display- graphic و fo: display-rule، وكذلك عناصر محولـــة مثل fo:inline-sequence و fo:page-number. قد تحتوي أيضا على نص خام. على ســبيل المثال:

```
<fo:block>
  <fo:inline-sequence font-style="italic">
   The XML Bible
  </fo:inline-sequence>
  Page <fo:page-number/>
  <fo:inline-sequence>
    Chapter 15: XSL Formatting Objects
  </fo:inline-sequence>
</fo:block>
```

يكون لدى عناصر fo:block سمات لكل من خصائص المساحة وخصائص تنسيق النـــص. يتم نقل خصائص تنسيق النص لأي عناصر تابعة في الكتلة إلا في حالة التجاوز تتضمن المتاحة:

- خصائص المحاذاة: text-align و text-align-last
- خصائص aural: azimuth, cue, cue-after, cue-before, elevation, pause, pause-after, pause-before, pitch, pitch-range, play-during, richness, speak, speak-header, speak-numeral, speak-punctuation, speech-rate, stress, voice-family و volume
- خصائص الخلفيـــة: background, background-attachment, background-color, background-image, background-position و background-repeat
- خصـــائص الحـــدود: border-before-color, border-before-style, border-before-width, border-after-color, border-after-style, border-after-width, border-start-color, border-start-style, border-start-width, border-end-color, border-end-style, border-end-width, border-top-color, border-top-style, border-top-width, border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-

left-style, border-left-width, border-right-color, border-right-style, border-right-width, border, border-top, border-bottom, border-left, border-width و border-right, border-color, border-style

- خصائص الفواصل: wrap-option و page-break-inside, widows, orphans
- خصائص اللون: color
- خصائص الأعمدة: span
- خصائص الخط: font-family, system-font, font-size, font-size-adjust, font و font-stretch, font-style, font-variant, font-weight
- خصائص الجملة المفصولة بخط :hyphenation country, hyphenate, hyphenation-char, hyphenation-push-char-count, hyphenation-remain-char-count, language, script, hyphenation-keep و hyphenation-ladder-count
- خصائص المسافات البادئة: last-line-end-indent و text-indent
- خصائص الطبقة: z-index
- خصائص طول الخط: line-height, line-height-shift-adjustment and, line-stacking-strategy
- خصائص الهامش: margin-bottom, margin-left, margin-right, margin-top, margin, margin, space-before, space-after, start-indent و end-indent
- خصائص Padding: padding-top, padding-bottom, padding-left, padding-right, padding-before, padding-after, padding-start و padding-end
- خصائص الموقع: left و position, top, bottom, right
- خصائص اتجاه النص: writing-mode
- خاصية الوضوح: visibility
- خصائص المسافات البيضاء: white-space-treatment

يعتبر معظم ما سبق من CSS سيتم مناقشته الباقي فيما بعد ولعناصر مستوى الكتلة الأخرى نفس قوائم الخاصية.

كائنات التنسيق الضمنية

ترسم كائن التنسيق الضمني كمساحة مستطيلة قد تحتوي على نص أو مساحات ضمنية أخرى. يتم ترتيب المساحات في أسطر من اليسار إلى اليمين عند امتلاء سطر، يتم بدء سطر جديد أسفل السطر السابق. مع ذلك يعتمد الترتيب الدقيق الذي يتم وضع العناصر الضمنية فيه على نمط الكتابة. على سبيل المثال، عند التعامل مع اللغة العربية أو العبرية، فمن المعقول وضع أول العناصر المحولة إلى اليسار ثم يتم الملء باتجاه جهة اليمين. تتضمن كائنات التنسيق الضمنية ما يلي:

- fo:bidi-override
- fo:character
- fo:first-line-marker
- fo:inline-graphic
- fo:inline-included-container
- fo:inline-rule
- fo:inline-sequence
- fo:list-item-body
- fo:list-item-label
- fo:page-number
- fo:page-number-citation

كائنات التنسيق الجدول

كائنات تنسيق الجدول التي تم تصميمها هو XSL المساوية لخصائص جدول CSS2. تعمل الجداول بصورة أكثر طبيعية في XSL أكثر منه في CSS. يعتبر الجدول الفردي كائن مستوى كتلة بينما أجزاء الجدول ليست كتلة محولة مع ذلك، يمكن لجدول كامل أن يتحول إلى كائن محول عن طريق Wrapping في fo:inline-included-container.

يوجد ٩ كائنات تنسيق جدول XSL:

- fo:table-and-caption
- fo:table
- fo:table-caption

- fo:table-column
- fo:table-header
- fo:table-footer
- fo:table-body
- fo:table-row
- fo:table-cell

جذر الجدول ليس fo:table ولكن fo:table-and-caption تحتوي على fo:table وfo:caption. يحتوي fo:table على fo:table-header وfo:table-body و-fo:table-footer يحتوي جسم الجدول على عناصر fo:table-row مقسمة إلى عناصر fo:table-cell.

كائنات تنسيق خارج السطر

هناك ثلاث كائنات تنسيق خارج الخط:

- fo:float
- fo:footnote
- fo:footnote-citation

تستعير كائنات التنسيق خارج الخط مساحة من كائنات الكتلة أو كائنات محولة موجودة بالفعل على الصفحة، ليس من الضروري أن يظهر بين العناصر نفسها التي ظهروا بينها في شجرة XML كائن تنسيق الإدخال.

قواعد

القاعدة هي خط أفقي مدرج في نص ولدى XSL نوعيين من الخطوط الأفقية يعتبر وكائن تنسيق fo:display-rule هو عنصر مستوى كتلة ينشئ خط أفقي مثل الذي ينتج بواسطة علامة <HR> الخاصة بلغة HTML. يتشابه عنصر كائن تنسيق fo:display-rule مع عنصر fo:inline-rule كما يوضح الاسم، فإن عنصر fo:inline-rule هو عنصر محول بدلاً من عنصر مستوى الكتلة ولهذا فهو يظهر في منتصف سطر النص ولا يوحي بفاصل خط على سبيل المثال، ما يلي هو قاعدة عرض.

مع ذلك، هذا ______________ هو قاعدة ضمنية

لكل من عناصر fo:inline-rule وfo:display-rule ست سمات أساسية تصفهم.

١- length: طول الخط 12pc أو 5in.

٢- rule-orientation: escapement, horizontal, line-progression, or vertical

٣- rule-style: تظل القيمة الفعلية ليتم تحديدها في وقت الكتابة.

٤- rule-thickness: سمك الخط، مثل 1px أو 0.1cm.

٥- vertical-align: baseline, bottom, middle, sub, super, text-bottom, text-top, top أو طول أو تحسين من ارتفاع الخط.

٦- color: لون خط، مثل pink و#FFCCCC.

على سبيل المثال، ما يلي هو قاعدة مستوى كتلة أخطر يكون طوله ٧,٥ بوصة وسمكه ٢ نقطة.

```
<fo:display-rule length="7.5in"
        line-thickness="2pt" color="#00FF00"/>
```

بالإضافة إلى ذلك، يمكن أن يكون لدى fo:display-rule معظم السمات المعتادة لعنصر مستوى الكتلة مثل الهوامش وPadding الموضحة ويمكن أن يكون لدى block-level السمات المعتادة لعنصر محول مثل fo:inline-rule الاستثناءات هي تلك السمات المرتبطة مباشرة بالنص مثل عائلة الخط وبالطبع هذه السمات ليست لا تكون ذات قيمة بالنسبة للقاعدة.

الرسومات

تتيح XSL أسلوبين لتضمين الصورة في مستند يدرج عنصر fo:display-graphic رسوم مستوى كتلة يدرج عنصر fo:inline-graphic رسوم محولة يتيح هذين العنصرين مثل علامة IMG اللغة HTML تصف الصورة ست سمات.

١- href: هو URI لملف الصورة

٢- min-height: هو الحد الأدنى لارتفاع الرأس للصورة

٣- min-width: هو الحد الأدنى للعرض الأفقي للصورة

٤- max-height: هو الحد الأقصى للارتفاع الرأسي للصورة

٥- max-width: هو الحد الأقصى للعرض الأفقي للصورة

٦- scale: مع قيمة max يقوم بتوسيع الرسم إلى حجم max-height وmax-width مع قيمة max-uniform يقوم بتوسيع الرسم بنفس مقدار الاتجاهات الأفقية والرأسية إما

إلى max-height أو max-width، أيهما يكون الأول، مع قيمة رقم فردي صحيح، قم بضرب كلاً من الارتفاع والعرض في هذا الرقم، ومع قيمة رقمين حقيقيين، قم بضـوب العرض في الأول والارتفاع في الثاني.

على سبيل المثال، لنفترض عنصر HTML IMG الخاص بلغة HTML القياسي:

```
<IMG SRC="logo.gif" WIDTH="100" HEIGHT="100"
    ALIGN="right" ALT="alt text" BORDER="0">
The fo:display-graphic element equivalent looks like this:
<fo:display-graphic image="logo.gif"
                height="100px" width="100px" />
```

الارتباط

بالنسبة للعروض المحولة فقط، يقدم XSL عنصر fo:simple-link بافتراض أنك فـي واجهـة مستخدم نمط مستعرض ويب. فإن النقر في أي مكان على في محتويات عنصر الارتباط يقفـز إلى الارتباط المستهدف. يعمل هذا العنصر كمستوى كتلة أو ارتباط محول بناء على ما يحتويـه. التحكم في أراء تصرف الارتباط الست سمات:

- الواجهة الخارجية
- الواجهة الداخلية
- الواجهة المشار إليها
- واجهة العرض
- كتلة وجهة المسافة فوق
- بداية وجهة المسافة فوق

يحدد الارتباط هدف مستند يعيد موقع URI عن طريق قيمة سـمة external-destination يجب تحميل المستند الموجود في موقع URI عند تنشيط الارتباط في بيئات GUI، يتـم تنشـيط الارتباط عن طريق النقر على محتويات الارتباط. على سبيل المثال:

```
<fo:block> Be sure to visit the
  <fo:simple-link
    external-destination="http://metalab.unc.edu/xml/">
     Cafe con Leche Web site!
  </fo:simple-link>
</fo:block>
```

يمكن أيضاً الارتباط لعقدة أخرى في نفس المستند باستخدام سـمة internal-destination قيمة هذه السمة ليست URI ولكن معرف العنصر الذي ستقوم بالارتباط إليه لا يجب تحديد كـلاً من الوجهة الخارجية والداخلية لهذا الارتباط.

تؤثر السمات الأربعة الأخرى فـي سـلوك ومظـهر الارتبـاط. لـدى سـمة indicate-destination قيمة Boolean "true أو false أو false افتراضياً" والتي تحدد ما إذا كان عندما يتم تحميل العنصر المرتبط إليه يجب تمييزه من الأجزاء غير المرتبط إليها في نفس المستند على سبيل المستند، إذا اتبعت ارتباط لعنصر ATOM واحد في جدول يحتوي علـى ١٠٠ ذرة مـن الممكن أن تكون الذرة المحددة التي سترتبط إليها بخط سميك بينما الذرات الأخرى تكون بخـط عادي وبقية التفاصيل تعتمد على النظام نفسه.

لدى سمات show-destination قيمتين محتملتين، replace وهي الافتراضية، وnew فـي حالة قيمة replace، عند اتباع ارتباط فإنه يستبدل المستند الحالي في نفس الإطار في حالة قيمـة new، عند اتباع المستند المستهدف في إطار جديد.

عندما يتبع مستعرض ارتباط HTML في منتصف مستند، يتم وضع عنصر الارتباط المحـدد فـي أعلـى الإطـار تتيـح سـمات space-above-destination-start وspace-above-destination-block تحديد أين يضع المستعرض العنصر المرتبط إليه إلى أسفل في الإطار عن طريق ترك مساحة محددة "ليست مساحة فارغة، ولكن غالباً ستحتوي على المحتويات التي تسبق العنصر المرتبط إليه" فوق عنصر الارتباط.

بالإضافة إلى ذلك قد يكون لدى الارتباط خاصية معتادة مثل اللون الذي سـيتم نقلـها إلـى محتويات الارتباط يتيح هذا تنسيق المحتويات الموجودة في الارتبـاط بصـورة مختلفـة عـن المحتويات الأخرى. على سبيل المثال، عن طريق وضع سطر أسفل كل الارتباط مـع ذلـك، لا تقدم كائنات تنسيق XSL وسيلة للتمييز بين الارتباطات التي تم زيادتها والتي لـم يتـم زيادتـها والارتباطات النشطة على عكس CSS وHTML.

القوائم

يصف عنصر كائن تنسيق fo:list-block عنصر قائمة مستوى الكتلة "لا توجد قوائم محولة" قـد تكون القائمة مرقمة، بها مسافات بادئة أو غير ذلك من وسائل التنسيق. كـل عنصـر fo:list-block يحتوي إما على سلسلة من عناصر fo:list-item أو العناصر الزوجيـة fo:list-item-label fo:list-item-body "لا يمكن أن يحتوي على الاثنين معاً" يجب أن تحتوي fo:list-item على fo:list-item-label وfo:list-item-body يحتـوي fo:list-item-label علـى علامـة نقطية، رقم أو أي علامة مميزة أخرى لعنـاصر القائمـة يحتـوي fo:list-item-body علـى

المحتوى الفعلي لعنصر القائمة للإيجاز، يحتوي fo:list-block على عناصر fo:list-item كـل عنصر fo:list-item يحتوي على fo:list-item-label وfo:list-item-body مع ذلك، يمكـن حذف عناصر fo:list-item. على سبيل المثال.

```
<fo:list-block>
  <fo:list-item>
    <fo:list-item-label>*</fo:list-item-label>
    <fo:list-item-body>Actinium</fo:list-item-body>
  </fo:list-item>
  <fo:list-item>
    <fo:list-item-label>*</fo:list-item-label>
    <fo:list-item-body>Aluminum</fo:list-item-body>
  </fo:list-item>
</fo:list-block>
```

أو، مع إزالة علامات fo:list-item:

```
<fo:list-block>
  <fo:list-item-label>*</fo:list-item-label>
  <fo:list-item-body>Actinium</fo:list-item-body>
  <fo:list-item-label>*</fo:list-item-label>
  <fo:list-item-body>Aluminum</fo:list-item-body>
</fo:list-block>
```

لدى عنصر fo:list-block ثلاث سمات خاصة:

١- provisional-label-separation: المسافة بين علامة عنصر القائمة ومحتوى عنصر القائمة يتم إعطائها كثلاثة هم الحد الأعلى والحد الأدنى والحد الأمثل: 0.5cm: 2cm: .Icm

٢- provisional-distance-between-starts: المسافة بين حافة البداية لعلامة عنصـر القائمة وحافة بداية محتوى عنصر القائمة.

٣- space-between-list-rows: المسافة الرأسية بين عناصر القائمة المتتالية التي تـم إعطائها ثلاثة هم الحد الأدنى والحد الأمثل، مثل 36pt;4pt;12pt.

لدى عنصر fo:list-item خصائص مستوى الكتلة القياسية للخلفيات والموقع والخط وفـاصل الصفحة والهوامش والحدود وغيرها.

الجداول

عنصر الجدول الأساسي في XSL هو fo:table-and-caption وهو كائن مستوى كتلة يمكن تحويلها إلى كائن محول عن طريق wrapping في fo:inline-included-container أو إلى كائن محول عن طريق wrapping في fo:float نموذج الجدول يتشابه مع HTML. يوضح الجدول ١٥-١ التشابه بين عناصر جدول HTML 4-0 وكائنات تنسيق XSL.

الجدول ١٥-١

جداول HTML مقارنة جداول كائنات تنسيق XSL

عنصر HTML	عنصر XSL Fo
TABLE	fo:table-and-caption
لا يوجد مثيل له	fo:table
CAPTION	fo:table-caption
COL	fo:table-column
COLGROUP	لا يوجد مثيل له
THEAD	fo:table-header
TBODY	fo:table-body
TFOOT	fo:table-footer
TD	fo:table-cell
TR	fo:table-row

يحتوي fo:table-and-caption على عنصر fo:caption اختياري وعنصر fo:table واحد يمكن أن يحتوي عنوان التسمية على أي عناصر مستوى كتلة تفضل أن يتم وصفه في عنوان التسمية افتراضياً، يتم وضع عناوين التسمية قبل الجدول ولكن يمكن تعديل ذلك عن طريق إعداد خاصية caption-side لعنصر table-and-caption لإحدى القيم الثمانية التالية:

- before
- after
- start
- end
- top
- bottom

- left ♦
- right ♦

على سبيل المثال، ما يلي هو جدول له عنوان تسمية في الأسفل:

```
<fo:table-and-caption caption-side="bottom">
  <fo:table-caption>
    <fo:block font-weight="bold"
            font-family="Helvetica, Arial, sans"
            font-size="12pt">
      Table 15-1: HTML Tables vs. XSL Formatting Object Tables
    </fo:block>
  </fo:table-caption>
  <fo:table>
    <!- table contents go here ->
  </fo:table>
</fo:table-and-caption>
```

يحتوي عنصر fo:table على عناصر fo:table-column, fo:table-header اختيارية. fo:table-footer, ينقسم إلى عناصر fo:table-body. كل fo:table-body يقسم إلى عناصر fo:table-row يمكن لعناصر fo:table-row أن تنقسم إلى عناصر fo:table-cell fo:table-header أو عناصر fo:table-footer أن تنقسم إلى fo:table-cell أو عناصر fo:table-row على سبيل المثال، ما يلي هو جدول بسيط يطابق أول ثلاثة صفوف لجدول ١٥-١.

```
<fo:table>
  <fo:table-header>
    <fo:table-cell>
      <fo:block font-family="Helvetica, Arial, sans"
              font-size="11pt" font-weight="bold">
        HTML Element
      </fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block font-family="Helvetica, Arial, sans"
              font-size="11pt" font-weight="bold">
        XSL FO Element
      </fo:block>
```

```
    </fo:table-cell>
   </fo:table-header>
   <fo:table-body>
    <fo:table-row>
     <fo:table-cell>
      <fo:block font-family="Courier, monospace">
       TABLE
      </fo:block>
     </fo:table-cell>
     <fo:table-cell>
      <fo:block font-family="Courier, monospace">
       fo:table-and-caption
      </fo:block>
     </fo:table-cell>
    </fo:table-row>
    <fo:table-row>
     <fo:table-cell>
      <fo:block>no equivalent</fo:block>
     </fo:table-cell>
     <fo:table-cell>
      <fo:block font-family="Courier, monospace">
       fo:table
      </fo:block>
     </fo:table-cell>
    </fo:table-row>
   </fo:table-body>
  </fo:table>
```

يمكن زيادة خلايا الجدول عدا صفوف وأعمدة عن طريق إعداد سمات n-columns-spanned و/أو n-rows-spanned على عدد صحيح وإعطاء عدد الصفوف والأعمدة المراد زيادتها سمة column-number الاختيارية يمكنها أن تغير أي عمود تبدأ الزيادة منه الافتراضي هو العمود الحالي.

يمكن رسم الحدود حول أجزاء الجدول باستخدام خصائص الحدود العادية التي ستناقش فيما بعد سمة column-number قيمة show أو hide أوshow إذا كانت الجدول سترسم حول الخلايا التي ليس لها محتويات وhide إذا لم تعطي. الافتراضي هوshow.

معظم أجزاء الجدول لا تستخدم خصائص العرض والارتفاع القياسية. بدلاً من ذلـــك، لـــهم سمات مماثلة. أي أو كل ذلك من الممكن حذفه أو في تلك الحالة يستطيع المنسق إعـــادة ضبــط حجم كل شيء.

- table-width, table-height :Table
- table-caption: الارتفاع يتم ضبطه آلياً بواسطة المنسق.
- table-row: العرض يتم تحديده بواسطة المحتوى.
- table-cell: column-number, cell-height عـــرض العمـــود n-columns-spanned, n-rows-spanned

لدى عنصـــر fo:table-row ســمات may-break-after-row و-may-break-before-row اختيارية لها قيم yes أو no لتحدد إذا كان مسموح بوضع فاصل صفحة قبل و بعد الصــف الافتراضي أن كليهما فيهما yes.

عند تحديد جدول عبر الصفحات المتعددة، يتم تكرار رأس وتذييل الصفحة في كــل صفحـــة يمكن تحديد هذا السلوك فـــي ســمات table-omit-middle-header لعنصـــر table-omit-middle-footer تشير قيمة fo: table أن رأس وتذييل الصفحة سيتم تكراره من صفحة إلـــى صفحة إلى الأخرى. تشير القيمة yes أن ذلك أن يحدث.الاقتراض هو no.

عنصر fo:table-column الافتراضي هو عنصر فارغ يحدد القيم لكل الخلايا في عمـــود محدد. الخلايا التي يتم تطبيقه عليها يتم تعريفها بواســـطة ســمة column-number عنصـــر fo:table-column لا يحتوي بالفعل على أي خلايا يمكن لعنصر fo:table-column أن يطبق خصائص لأكثر من عمود متتالي عن طريق إعداد خاصية n-columns-spanned على عـــدد صحيح أكبر من واحد. الخاصية الأكثر شيوعاً التي يتم إعدادها فـــي fo:table-column هـــي column-width "طول موقع" ولكن خصائص الحدود القياسية، الخلفية Padding "ستناقش فيمــا بعد" يمكن إعدادها أيضاً.

الأحرف

يستبدل كائن تنسيق fo:character حرف محدد أو سلسلة من الأحرف في المدخلات مع حــرف مختلف في المخرجات يمكن استخدام هذه الترجمة بين النقطة العشـــرية الأمريكيـــة والفاصلـــة العشرية الأمريكية، على سبيل المثال، تحدد سمة character أي حرف استبدال يتم اســـتخدامه. على سبيل المثال، تستبدل قاعدة القالب هذه للأحرف في عنصر PASSWORD.

```
<xsl:template match="PASSWORD">
  <fo:character character="*">
```

```
<xsl:value-of select="."/>
</fo:character>
</xsl:template>
```

مع ذلك، يعتبر هذا الاستخدام نادر الهدف الأساسي لعنصر fo:character هو أن محركات التنسيق يمكن أن تعالج كل حرف على أن الحرف الخاص بها إذا كنت تكتب محرك تنسيق يمكن تجاهل هذا العنصر.

التتالي

لا يوجد تأثير محدد للتتالي على المخططات سواء المحولة أو مربعات مستوى الكتلة هم ببساطة عناصر يمكنك تعليق سمات التنسيق قبل font-style أو text-indent التطبيق لتوابع التتالي:

يعتبر عنصر كائن تنسيق fo:display-sequence حاوية تجمع كائنات مستوى الكتلة معاً. يمكن فقط جمع عناصر مستوى الكتلة مثل fo:display-graphic وfo:block لا يمكنها أن تحتوي على عناصر محولة أو نص هام.

يعتبر عنصر كائن تنسيق fo:inline-sequence حاوية تجمع الكائنات المحولة معاً. لا يمكن أن تحتوي على عناصر مستوى كتلة على سبيل المثال، يمكن استخدام عناصر inline-sequence لإضافة نمط للأجزاء مختلفة لتذييل الصفحة قبل:

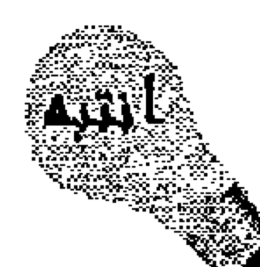
```
<fo:flow id="q2" flow-name="xsl-after">
  <fo:block font-style="bold" font-size="10pt"
        font-family="Arial, Helvetica, sans">
    <fo:inline-sequence font-style="italic"
                text-align="start">
     The XML Bible
    </fo:inline-sequence>
    <fo:inline-sequence text-align="centered">
     Page <fo:page-number/>
    </fo:inline-sequence>
    <fo:inline-sequence text-align="right">
     Chapter 15: XSL Formatting Objects
    </fo:inline-sequence>
  </fo:block>
</fo:flow>
```

الحاشية السفلية

يمثل عنصر fo:footnote حاشية سفلية يضع المؤلف عنصر fo:footnote في تدفق حيث موضع مرجع حدوث الحاشية السفلية قبل fo:footnote-reference يحتوي عنصر fo:footnote على كل من fo:footnote-reference وعنصر مستوى كتلة يحتوى على نص الحاشية السفلية. ومع ذلك يتم فقط إدراج مرجع الحاشية السفلية بطريقة محولة. يضع المنسق نص الملحوظة في منطقة ما بعد وهي تذييل الصفحة.

على سبيل المثال، تستخدم الحاشية السفلية العلامة النجمية كعلامة حاشية سفلية إلى " JavaBeans, Elliotte Rusty Harold (IDG Books, Foster City, 1998), p. 147". خصائص XSL القياسية قبل JavaBeans وElliotte Rusty Harold يتم استخدامها لتنسيق كلاً من علامة الملاحظة والنص بالأسلوب التقليدي.

```
<fo:footnote>
  <fo:footnote-reference
   font-size="smaller" vertical-align="super">
   *
  </fo:footnote-reference>
  <fo:block font-size="smaller">
    <fo:inline-sequence
     font-size="smaller" vertical-align="super">
      *
    </fo:inline-sequence>
    <fo:inline-sequence
     font-style="italic">JavaBeans</fo:inline-sequence>,
     Elliotte Rusty Harold
     (IDG Books, Foster City, 1998), p. 147
  </fo:block>
</fo:footnote>
```



لا تقدم مفردات كائنات التنسيق أي وسائل للترقيم ونقل الحواشي السفلية آلياً، ولكن يمكن قبل ذلك باستخدام xsl:number في ورقة نمط التحويل كذلك يسهل استخدام تحويلات XSL عمل ملاحظات النهاية.

التنقل

ينتج عن fo:float مربع تنقل يكون إرسائه على المنطقة التي يقع فيها. مستخدم fo:float فـــي الرسومات، التخطيطات، والجداول أو أي محتويات خارج الخط. تحتاج أن تظهر في مكان مـــا في الصفحة ولكن أين ستظهر. ليس مهماً بالمرة. على سبيل المثال، ما يلي هو رمز لرســومات التنقل مع عنوان تسمية مضمن في وسط الفقرة.

```
<fo:block>
 Although PDF files are themselves ASCII text,
 this isn't a book about PostScript, so there's
 nothing to be gained by showing you the exact
 output of the above command. If you're curious,
 open the PDF file in any text editor.
 Instead, Figure 15-1
 <fo:float>
   <fo:display-graphic
    image="3236-7fg1501.jpg"
    height="485px" width="623px" />
   <fo:block font-family="Helvetica, sans">
    <fo:inline-sequence font-weight="bold">
      Figure 15-1:
    </fo:inline-sequence>
    The PDF file displayed in Netscape Navigator
   </fo:block>
 </fo:float>
 shows the rendered file displayed in
 Netscape Navigator using the Acrobat plug-in.
</fo:block>
```

يبذل المنسق أقصى جهد لوضع الرسوم في مكان ما على نفس الصفحة بينما المحتويات التـي تحيط عنصر fo:float تظهر، على الرغم أن هذا غير متاح دائماً، في تلك الحالة فإنـــه يحـــرك الكائن للصفحة التالية. في تلك الحدود، من الممكن أن تضعها في أي مكان.

خصائص تنسيق XSL

لا تذكر كائنات التنسيق الكثير بشأن كيفية تنسيق المحتوى، فهم يضعون محتويات في مربعـــات يتم وضعها أجزاء محددة من الصفحة. تحدد السمات بالنسبة لكائنات التنسيق المختلفة كيف يتـــم وضع نمط المحتويات في هذه المربعات.

كما ذكرنا من قبل، يوجد على الأقل ٢٠٠ خاصية تنسيق منفصل لا يمكن للخصائص كلها أن تتم ربطها لكل العناصر على سبيل المثال، لا يوجد هـــدف مـــن تحديـــد font-style لعنصـــر fo:display-graphic معظم الخصائص، يمكن تطبيقها لأكثر من نوع عنصـــر كـــائن تنســـيق بعضهم لا يستطيع، مثل href وprovisional-label-separation، تم مناقشتها فيما سبق مـــع كائنات التنسيق المطبقين عليهم عندما تكون خاصية شائعة لعدة كائنات تنسيق فإنها تشارك نفـــس بناء الجملة والمعنى عبر الكائنات. على سبيل المثال، يمكن استخدام رمـــوز متطابقـــة لتنســـيق fo:list-label في خط سميك يكون ١٤ نقطة fo: list-label كما تفعل في تنسيق fo:block في الخط السميك ١٤ نقطة fo:block.

تتشابه العديد من خصائص XSL مع خصائص CSS قيمة خاصية CSS font-family فـــي CSS هي نفسها كقيمة سمة font-family في XSL إذا كنت قرأت من CSS فـــي الفصـــل ١٢ و١٣ فقد تعلمت معظم ما تحتاجه في خصائص XSL.

أنواع البيانات والوحدات

قد تكون قيمة خاصية تنسيق XSL كلمة أساسية مثل auto, italic أو transparent أو قد تكون قيمة حرفية مثل true, 5px, -5.0cm أو http://www.w3.org/index.html تعطي القيـــم الحرفية في XSL كواحد من ٢٤ نوع بيانات وهي مدرجة في جدول ١٥-٢.

الجدول ١٥-٢
أنواع بيانات خاصية التنسيق

نوع البيانات	التعريف	أمثلة
Name	رمز اسم XML فريد	q1 copyright
ID	مفهوم اسم يماثل معرف عنصـــر في المستند	q1 copyright
IDREF	أمـــا سلســـلة 'True' أو سلســـلة 'False'	q1 Copyright

الجدول ١٥-٢

أنواع بيانات خاصية التنسيق

نوع البيانات	التعريف	أمثلة
Boolean	حرف UniCod منفرد، دون مساحة بيضاء	True False
Char	حرف Unicode منفرد، دون مساحة بيضاء	A _
Signed Integer	تسلسل من الأرقام، يتم اختيار سقها إما بعلامة جمع (+) أو علامة طرح (-)	0 -28 +1000000000
Unsigned Integer	تسلسل الأرقام	0 28 1000000000
Positive Integer	تسلسل أرقام يتضمن على الأقل محدد واحد ليس صفر	28 1000000000
Signed Real	رقم نقطة عام فى أحرف نقطة أحرف علامة التنسيق الرموز Exponential لا يتم تدعيمها علامة (+) هى اختيارية للأرقام الموجبة	+0.879 -31.14 2.71828
Unsigned Real	رقم نقطة بما تم غير سلبي في أحرف نقطة أحرف التنسيق لا يتم تدعيم الرموز Exponential	0.0 31.14 2.71828
Positive Real	عدد صحيح موقع أو حقيقى موقع يتبعه وحدة	0.01 31.14 2.71828
Signed Length	عدد صحيح غير موقع أو حقيقي رقم غير موقع يتبعه بوحدة	5px -0.5in
Unsigned Length	عدد صحيح موقع يجب تقسيمه على ١٠٠ للحصول على القيمة الفعلية	10px 0.5cm
Percent	الحد الأدنى لطول الفاصلة المنقوطة، الحد الأقصى	100.0 -43.2 0.0

الجدول ١٥-٢

أنواع بيانات خاصية التنسيق

نوع البيانات	التعريف	أمثلة
Space Specifier	لطول الفاصلة المنقوطة. الطول الأمثل الفاصلة المنقوطة، ما يسبق الفاصلة المنقوطة بشروط	0px;72px;12px; force;discard
Limit Specifier	الحد الأدنى لطول الفاصلة المنقوطة، الحد الأقصى للطول	0px;72px
	لون مسمى أو ثلاثي سداسي عشري في نموذج #RRGGBB	White #FFFFFF
URI	محدد موارد عام، عليا يسمى URL	Http://www.w3 org/index.html /index.html / ../index.html
Language	رمز لغة ISO 639	En La
Font Name	اسم الخط سواء فعلي أو رمزي	Times New Roman Serif
Font List	أسماء الخطوط مفصول بواسطة فاصلة وغالبا مسافات بيضاء	Times New Roman, Times, serif
Enumeration	ترقيم XML	(airplane | train | car | horse)
String	أي تتالي للأحرف	Fred Lucy and Ethel Castles don't have phones

خصائص إخبارية

يوجد خاصيتان إخباريتان يمكن تطبيقهما على أي كائن تنسيق. ومع ذلك، ليس لدى أي منهما تأثير مباشر على التنسيق. ولكن لا يعتبر أي منهما خصائص تنسيق.

خاصية id

أول خاصية هي id وهي سمة نوع id لـ XML وقيمة هذه الخاصية يجب أن تكون اسم XML الوحيد في ورقة النمط وفي مستند كائن تنسيق المخرجات. قد ينتج عن قاعدة القالب الموجود في

ورقة النمط مئات من العناصر في مستند المخرجات. يمكن الاستفادة هنا من دالـــة generate-id() لتحويلات XSL.

خاصية اللغة

تعتبر ثاني خاصية هي language وهذا يحدد اللغة من المحتوى الموجود في هـــذا العنصـــر. وعموماً فإن قيمة هذه الخاصية هي كود لغة ISO 639 مثـــل (English) أو (Latin) وربمـــا تكون الكلمة الأساسية none أو use-document. ويعني الأخير ببساطة استخدام لغة الإدخـــال كما تم تحديدها بواسطة سمة xml:lang فمثلاً اعتبر المقطـــع الأول مـــن Caesar's Gallic Wars:

```
<fo:block id="verse1.1.1" language="la">
  Gallia est omnis divisa in partes tres,
  quarum unam incolunt Belgae, aliam Aquitani,
  tertiam qui ipsorum lingua Celtae, nostra Galli appellantur
</fo:block>
```

بالرغم من أن خاصية language ليس لها أي تأثير على التنسيق ربما يكون لديها تأثير غير مباشر إذا حدد المنسق حسابات تخطيط معتمداً على اللغة. فمثلاً ربما يستخدم المنسق طرق كتابة إضافية مختلفة عن النص العربي والنص الإنجليزي.

خصائص الفقرة

خصائص الفقرة هي أنماط لتطبيق كتلة من نص كامل في معالج النصوص التقليدي، بالرغم مـــن أن خصائص نص مستوى الكتلة هي الأنسب هنا. فمثلاً تعتبر المسافات البادئة هي خاصية الفقرة لأنه يمكنك ترك فراغ في فقرة لكن لا تستطيع فعل ذلك في كلمة واحدة منفصلة عن فقرتها.

خصائص الفواصل

تحدد خصائص الفواصل أين يمكن السماح بوضع فواصل صفحات وأين لا يسمح بوضعها وهـــي ليست مرتبطة بخصائص الفواصل.

- keep-with-next
- keep-with-previous
- break-before
- break-after

- inhibit-line-breaks

تعتبر كل من خاصتي keep-with-next وkeep-with-previous همــا Booleans مــع تحديد ما إذا كان يجب لكائن التنسيق أن يظل في نفس كائن التنسيق الرئيسي كالتنســيق الســابق واللاحق. وهذا يؤثر على الحفاظ على كائني التنسيق في نفس الصفحة ولكنها أكثر حزمــاً مــن ذلك:

تدرج خاصية break-before فاصل قبل أن يبدأ كائن التنسيق. تتضمن الأشــياء الممكنــة للعوامل column وpage وodd-page وeven-page وربما تكــون none أو auto-page. وتدرج خاصية break-after فاصل بعد أن ينتهي كائن التنسيق. يتم استخدام نفس القيم break-before. فمثلاً تؤكد قاعدة القوالب أن كل SONNET ذات حجم صغير تقوم بالطباعة في صفحة خاصة بها.

```
<xsl:template match="SONNET">
  <fo:block break-before="page" break-after="page">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

أخيراً فإن خاصية inhibit-line-breaks هي Boolean التي يمكن إعدادها للإشارة إلى أنــه ليس هناك فاصل مسموح.

خصائص الواصلة

تحدد خصائص الواصلة ما إذا كان مسموحاً بها وكيف يتم استخدامها. يتم تطبيق هذا فقط علــى الواصلات الاختيارية مثل تلك التي يتم استخدامها لفصل الكلمات الطويلة في نهاية السطر. ولكن لا يتم تطبيقها في كلمة mother-in-law بالرغم من أنها قد تؤثر في مكان إتاحــة الواصــلات الاختيارية. وهناك ستة خصائص للواصلات هي:

- hyphenate: واصلات آلية يسمح بها فقط إذا كانت لــدى خاصيــة Boolean قيمــة true.
- hyphenation-char: يتم استخدام حرف Unicode لوصــل الكلمــات مثــل – فــي الإنجليزية.
- hyphenation-keep: أحد الأربــع كلمــات الأساســية "column وnone وpage, وspread" التي تحدد ما إذا كان يمكن استخدام الواصلة في نهاية صفحات متقابلــة أو عمود.
- hyphenation-ladder-count: رقم غير معين يحدد العدد الأكبر من الأسطر التي يتم تطبيق الواصلة عليها والتي قد تظهر في صف.

- hyphenation-push-char-count: رقم غير معين يحدد العدد الأصغر من الأحرف المتبوعة آلياً بواصلة. "تبدو المقاطع القصيرة سيئة وهي منفصلة".
- hyphenation-remain-char-count: رقم غير معين يحدد العدد الأصغر من الأحرف المسبوقة آلياً بواصلة.

يمكن اعتماد الواصلة على اللغة والنص المستخدم لذا فإن الخصائص الثلاثة لديها تأثير معين هنا:

- country
- language
- script

على سبيل المثال:

```
<fo:block hyphenate=true
        hyphenation-char="-"
        hyphenation-keep="none"
        hyphenation-ladder-count="2"
        hyphenation-push-char-count="4"
        hyphenation-remain-char-count="4" >
  some content...
</fo:block>
```

لا يحدد XSL حساب فصل مقطع لتحديد أين يتم تطبيق واصلة اختيارية. حتى باستخدام هذه الخصائص التي توفر الواصلات، فإنه لا يزال متاحاً أمام للمنسق كيفية وصل الكلمات ال منفردة.

خاصية المحاذاة العمودية

تحدد خاصية vertical-align الموضع العمودي لكائن التنسيق في السطر. وهي تؤدي نفس وظيفة خاصية CSS2 لنفس الاسم. يوجد ثمان قيم كلمات أساسية لهذه الخاصية:

١- baseline: تقوم بمحاذاة الخط الأساسي للمربع بالخط الأساسي لمربع الخط.

٢- Sub: تقوم بمحاذاة الخط الأساسي للمربع بالخط الأساسي لقيم الارتفاع الموجودة داخل مربع الخط.

٣- Super: تقوم برفع الخط الأساسي للمربع إلى الخط الأساسي لقيم الارتفاع الموجودة في مربع الخط.

٤- Top: تقوم بمحاذاة أعلى المربع بأعلى مربع الخط.

٥- Middle: تقوم بمحاذاة النقطة الوسطى بالخط الأساسي لمربع الخط بالإضافة إلى نصف ارتفاع x لمربع الخط.

٦- أسفل: تقوم بمحاذاة أسفل المربع بأسفل مربع الخط.

٧- أعلى النص: تقوم بمحاذاة أعلى المربع بأعلى الخط.

٨- أسفل النص: تقوم بمحاذاة أسفل المربع بأسفل الخط.

يمكنك أيضاً تعيين محاذاة عمودية إلى طول معين يقوم برفع أو تخفيض المربع بواسطة المسافة المحددة من الخط الأساسي.

خصائص المسافة البادئة

تحدد خصائص المسافة البادئة الأربع التالية start-indent وend-indent وtext-indent وlast-line-end-indent بمدى إضافة مسافة بادئة للأسطر من حاشية النص. تقوم خاصية start-indent بإزاحة جميع الأسطر من حاشية البدء "أيسر الحاشية في الإنجليزية". تقوم خاصية end-indent بإزاحة السطر الأول فقط من حاشية البدء. تقوم خاصية last-line-end-indent بإزاحة السطر الأخير فقط من حاشية البدء. يتم إتاحة قيم كطول مميز. يؤدي استخدام قيمة موجبة بالنسبة إلى start-indent وقيمة سالبة بالنسبة إلى text-indent إلى تكوين مسافات معلقة. على سبيل المثال، قد يتم تنسيق فقرة قياسية تحتوي على 0.5 بوصة مسافة البادئة للسطر الأول كما يلي:

```
<fo:block text-indent="5.0in">
The first line of this paragraph is indented
</fo:block>
```

يتم تنسيق علامة اقتباس كتلة تحتوي على 1 بوصة لبادئة المسافة في كل الأسطر على كلا الجانبين كما يلي:

```
<fo:block start-indent="1.0in" end-indent="1.0in">
  This text is offset one inch from both edges.
</fo:block>
```

خصائص الأحرف

توضح خصائص الأحرف صفات الأحرف الفردية، بالرغم من إمكانية تطبيقها على العناصر التي تحتوي على أحرف مثل عنصري fo:block وfo:list-item-body . يتضمن ذلك اللون والخط والنمط والوزن وخصائص متشابهة.

خاصية اللون

تقوم خاصية color بتحديد اللون الأمامي للمحتويات باستخدام نفس بناء الجملة كما في خاصية CSS color. على سبيل المثال، يؤدي ذلك إلى تلوين النص "Lions and tigers and bears, oh my! بلون pink":

```
<fo:inline-sequence color="#FFCCCC">
  Lions and tigers and bears, oh my!
</fo:inline-sequence>
```

خصائص الخط

يحتوي أي كائن تنسيق يتولى النص على نطاق واسع من خصائص الخط المتعارف عليها من خلال CSS، ومنها ما يلي:

- font-family: قائمة بأسماء الخط من حيث التفضيل.
- font-size: طول بعلامة.
- font-size-adjust: النسبة المفضلة بين ارتفاع X وحجم الخط محددة كرقم حقيقي بدون علامة أو none.
- font-stretch: عرض الخط كأحد الكلمات الأساسية التالية: condensed أو expanded أو extra-condensed أو extra-expanded أو narrower أو normal أو semi-condensed أو semi-expanded أو ultra-condensed أو ultra-expanded أو wider.
- font-style: أسلوب الخط محدد كأحد الكلمات الأساسية التالية italic أو normal أو oblique أو reverse-normal أو reverse-oblique.
- font-variant: إما normal أو small-caps.
- font-weight: سمك ضغط المفاتيح التي ترسم الخط متاح كأحد الكلمات الأساسية التالية 100 و200 و300 و400 و500 و600 و700 و800 و900 وbold وbolder وlighter وnormal.

خاصية تحويل النص

تحدد خاصية text-transform كيفية كتابة النص بالأحرف اللاتينية الكبيرة ومطابقته لخاصية css بنفس الاسم. وفيما يلي الأربع قيم المحتملة:

- none: لا تغير الحالة "الافتراضي".

- Capitalize: تجعل أول حرف من الكلمة حرف لاتيني كبير وبقيـــة الأحـــرف التاليـــة أحرف لاتيني صغيرة.
- Uppercase: تجعل جميع الأحرف لاتيني كبيرة.
- Lowercase: تجعل جميع الأحرف لاتيني صغيرة.

تتعلق هذه الخاصية باللغة. فمثلاً، اللغة الصينية لا تحتوي على حالات أحرف كبيرة وصغيرة منفصلة. تتحرر التنسيقات من حالات الأحرف عند تطبيقها على نص غير النص اللاتيني.

خاصية تظليل النص

تطبق خاصية text-shadow تظليل على النص. يتشابه ذلك مع لون الخلفية ويختلف في ارتباط الظل "التظليل" بالنص نفسه بدلاً من المربع الذي يحتوي النص. قد تكون قيمـــة text-shadow هي الكلمة الأساسية none أو a named or RGB color. مثل:

```
<fo:inline-seqence text-shadow="FFFF66">
  This sentence is yellow.
</fo:inline-sequence>
```

خاصية زخرفة النص

تعتبر خاصية text-decoration مطابقة لخاصية زخرفة النص CSS2. تحتوي على الخمس قيم المحتملة التالية:

- none
- underline
- overline
- line-through
- blink

الافتراضي لا شيء

خاصية تباعد النقاط

تعتبر النقاط كلمة شاملة التسطير أسفل الكلمـــة **underlining** ويتوســطه خــط linthrough ويتوسطه خط مزدوج (double strike-through) وهكذا. تحدد خاصيـــة score-spaces،إذا كان يوجد تسطير أسفل المسافة البيضاء. على سبيل المثـــال، إذا كـــانت score-spaces هـــي true،ثنن تبدو الكلمة الموجودة بأسفلها نظير هكذا an underlined sentence looks like this. وإذا كانت score-spaces هي false، تبدو الكلمة الموجود بأسفلها تســطير هكـــذا an underlined sentence looks like this.

خصائص الجمل

يتم تطبيق خصائص الجمل على مجموعات من الحرف وهي خاصية ذات معنى عندما يتم تطبيقها على أكثر من حرف في وقت واحد مثل المسافة بين الأحرف أو الكلمات.

خصائص المسافة بين الأحرف

لا يوجد عدد محدد للمسافات التي تفضل بين حرفين، فمعظم التنسيقات تقوم بضبط المسافة بين الأحرف بناءاً على الضرورة الموضعية وخاصة في النص المضبوط. بالإضافة إلى ذلك، تستخدم الخطوط عالية الجودة كميات مختلفة من المسافة الموجودة بين glyphs. ومع ذلك، يمكنك التحكم في ضيق أو اتساع النص بأكمله.

تدرج خاصية letter-spacing مسافة إضافية بين كل pair of glyphs وهي متاحة كطول مميز يحدد الكمية المطلوبة من المسافات الزائدة المراد إضافتها على سبيل المثال:

```
<fo:block letter-spacing="1.5px">
  This is fairly loose text
</fo:block>
```

يمكنك تحديد الطول بسالب لتضبيق النص. ولكن ستضع التنسيقات حدود على كمية المسافات الزائدة المتاح بإضافتها إلى أو إزالتها من المسافة بين الأحرف.

خصائص المسافة بين الكلمات

تقوم هذه الخاصية بضبط كمية المسافة بين الكلمات وإلا ستؤدي نفس وظائف خصائص المسافة بين الأحرف. تعتبر القيمة هي طول مميز يعطي كمية المسافة الزائدة المراد إضافتها بين كلمتين. مثل:

```
<fo:block word-spacing="3.0cm">
  This is pretty loose text.
</fo:block>
```

خصائص المسافة بين الأسطر

يقوم محرك تنسيقات XSL بتقسيم مساحة كتلة إلى مساحة أسطر. لا يمكنك إنشاء مساحة سطر مباشرة من XSL. ومع ذلك، باستخدام الخصائص الخمس التالية يمكنك التأثير في كيفية وضع المسافات عمودياً:

- line-height: الارتفاع الأدنى للسطر.

- line-height-shift-adjustment: consider-shifts إذا كان يجب على كـل مـن subscripts و superscripts. توسيع ارتفاع السطر disregard-shifts إذا لم يكـن فرض.
- line-stacking-strategy: line-height "نموذج CSS والافتراضي" font-height ; "يجعل السطر مرتفع كارتفاع الخط بعد إضافـة" font-height-override-before أو font-height-override-after "المسافة بين أقصى ارتفاع متزايد وأقصى عمق متناقص".
- font-height-override-after: طول مميز يحدد مسافة عمودية إضافية يتم وضعها بعد كل سطر، وقد تكون أيضاً الكلمة الأساسية use-font-metrics (الافتراضي) للإشارة إلى الاستناد إلى الخط.
- font-height-override-before:طول مميز يحدد أصغر مسافة عمودية إضافية يتم وضعها قبل كل سطر، وقد تكون أيضاً الكلمة الأساسية use-font-metrics "الافتراضي" للإشارة إلى الاستناد إلى الخط.

يعتمد ارتفاع السطر أيضاً على حجم الخط الذي يرسم به السطر فأحجام الخط الكبيرة سوف تحتوي بالتأكيد على أسطر طويلة. على سبيل المثال، تلك الفقرة الافتتاحية المأخوذة من Mary Wollstonecraft's Of the Rights of Woman والتي يوجد بها مسافة مزدوجة:

```
<fo:block font-size="12pt" line-height="24pt">
  In the present state of society it appears necessary to go
  back to first principles in search of the most simple truths,
  and to dispute with some prevailing prejudice every inch of
  ground. To clear my way, I must be allowed to ask some plain
  questions, and the answers will probably appear as
  unequivocal as the axioms on which reasoning is built;
  though, when entangled with various motives of action, they
  are formally contradicted, either by the words or conduct
  of men.
</fo:block>
```

خصائص محاذاة النص

تحدد كل من خاصتي text-align و text-align-last كيفية محاذاة المحتوى الداخلي أفقياً مع المربع الذي يوجد به تعتبر القيم الست المحتملة هي:

١- start: محاذاة إلى اليسار في النصوص من اليمين إلى اليسار.

٢- Centered: في الوسط.

٣- End: محاذاة إلى اليمين في النصوص من اليمين إلى اليسار.

٤- Justify: تتسع بمسافة زائدة ضرورية لملء السطر وهي الحاشية اليسرى في الصفحة اليسرى لصفحتين متقابلتين أو الحاشية اليسرى في الصفحة اليمنى لصفحتين متقابلتين.

٥- page-inside: محاذاة حاشية الصفحة وهي الحاشية اليسرى في الصفحة اليسرى لصفحتين اليمنى لصفحتين متقابلتين.

٦- page-outside: محاذاة مع الحاشية الخارجية للصفحة وهي الحاشية اليسرى في الصفحة اليسرى لصفحتين متقابلتين أو الحاشية اليمنى في الصفحة اليمنى لصفحتين متقابلتين.

تمكنك خاصية text-align-last من تحديد قيمة مختلفة لأطر سطر في النص. يعتبر ذلك ذو أهمية كبيرة للنص المضبوط، حيث لا يحتوي السطر الأخير عادة على كلمات كافية لضبطه. تعتبر القيم المحتملة هي ما يلي start, end, justified وrelative تستخدم القيمة relative نفس القيمة مثل خاصية text-align إلا إذا كانت text-align هي justified في أي الحالتين سيتم محاذاة السطر الأخير مع حاشية البداية.

خاصية المسافة:Whitespace

تحدد خاصية whitespace تصرف محرك التنسيق مع المسافة الموجودة بعد نقل مستند المصدر الأصلي إلى كائنات التنسيق. توجد ثلاثة قيم محتملة هي:

١- preserve: تترك المسافة كما هي.

٢- collapse: تضغط كل المسافات البيضاء إلى مسافة واحدة.

٣- ignore: يحذف أي مسافة بيضاء أولى أو أخيرة

ولكن من الأفضل الإبقاء على أي مسافة بعد التحويل. حيث أنها إذا لم تكن ذات أهمية، يسهل على عملية التحويل التخلص منها باستخدام xsl:strip.

خاصية خيار الالتفاف

تحدد خاصية كيفية التعامل مع نص طويل للغاية ليتناسب مع سطر واحد، تحتوي هذه الخاصية على قيمتي كلمتان أساسيتان وهما:

١- wrap: التفاف سهل للنص إلى السطر التالي.

٢- no-wrap: لا يحدث التفاف للنص.

خصائص المساحة

يتم تطبيق خصائص المساحة على المربعات. وقد تكون هـــذه الخصـــائص. مســتوى كتلـــة أو مربعات داخلية. تحتوي كل هذه المربعات على:

- خلفية
- هوامش
- حدود
- حشو
- حجم

خصائص الخلفية

تتطابق خصائص الخلفية مع خصائص خلفية CSS1. وهي خمس خصائص:

- تحدد خاصية background-color لون خلفية المربع. تعتبر قيمتها هي لون أو الكلمـــة الأساسية transparent.
- توفر خاصية background-image URI لصورة سيتم استخدامها كخلفية. قد تكـــون القيمة هي الكلمة الأساسية none.
- تحدد خاصية background-attachment ما إذا كان سيتم إرفاق الخلقية بالإطـــار أو بالمستند. وقيمتها هي أحد الكلمتان الأساسيتان fixed أو scroll..
- تحدد خاصية background كيفية وضع صورة خلفية في مربع تتضمن القيم المحتملـــة ما يلي: center, left, right أو bottom أو middle أوtop.
- تحدد خاصية background-repeat كيفية تغطية الصورة للمربع إذا كـــانت خاصيـــة repeat وno-repeat وrepeat-x وrepeat-y.

توضح الكتلة التاليـــة اســتخدام كـــل مـــن الخصـــائص التاليـــة: background-image وbackground-position و background-repeat وbackground-color.

```
<fo:block background-image="/bg/paper.gif"
        background-position="0,0"
        background-repeat="repeat"
        background-color="white">
  Two strings walk into a bar...
</fo:block>
```

خصائص الحدود

توضح خصائص الحدود مظهر الحد حول المربع وعادة ما تكون هي نفسها كخصائص حدود CSS ومع ذلك، تحتوي إصدارات XSL أيضاً على إصدارات border-XXX-before و border-XXX-after و border-XXX-start و border-XXX-end and بالإضافة إلى border-XXX-bottom و border-XXX-top و border-XXX-left و border-XXX-right.

يوجد ٣١ خاصية حدود وهي:

- اللون: border-color, border-before-color, border-after-color, border-start-color, border-end-color, border-top-color, border-bottom-color, border-left-color, border-right-color. لون الحد الافتراضي هو أسود.
- العرض: border-width, border-before-width, border-after-width, border-start-width, border-end-width, border-top-width, border-bottom-width, border-left-width, border-right-width.
- الأسلوب: border-style, border-before-style, border-after-style, border-start-style, border-end-style, border-top-style, border-bottom-style, border-left-style, border-right-style
- خصائص الاختزال: border, border-top, border-bottom, border-left, border-right, border-color, border-style, border-width

على سبيل المثال، يؤدي ذلك إلى رسم مربع أزرق يبلغ اتساعه ٢ بكسل حول الكتلة:

```
<fo:block border-before-color="blue" border-before-width="2px"
          border-after-color="blue" border-after-width="2px"
          border-start-color="blue" border-start-width="2px"
          border-end-color="blue" border-end-width="2px">
  Two strings walk into a bar...
</fo:block>
```

خصائص الحشو

تحدد خصائص الحشو بين حدود المربع ومحتوياته. تقع حدود المربع إذا تم توضيحها بين الهامش والحشو. تعتبر خصائص الحشو هي نفس خصائص حشو CSS. ومع ذلك، بالإضافة إلى خصائص padding-bottom و padding-top و padding-left و padding-right تحتوي أيضاً إصدارات XSL على إصدارات padding-before و padding-after و padding-start

وand padding-end. وبالتالي يبلغ مجموع خصائص الحشو ثمانية خصائص، يحتوي كل منها على طول مميز لقيمة ما. وفيما يلي هذه الخصائص:

- padding-after
- padding-before
- padding-bottom
- padding-end
- padding-left
- padding-start
- padding-right
- padding-top

وعلى سبيل المثال، تحتوي هذه الكتلة على حشو يساوي ٠,٥ سم على جميع الجوانب:

```
<fo:block padding-before="0.5cm" padding-after="0.5cm"
        padding-start="0.5cm"  padding-end="0.5cm">
  Two strings walk into a bar...
</fo:block>
```

خصائص الهامش للمجموعات

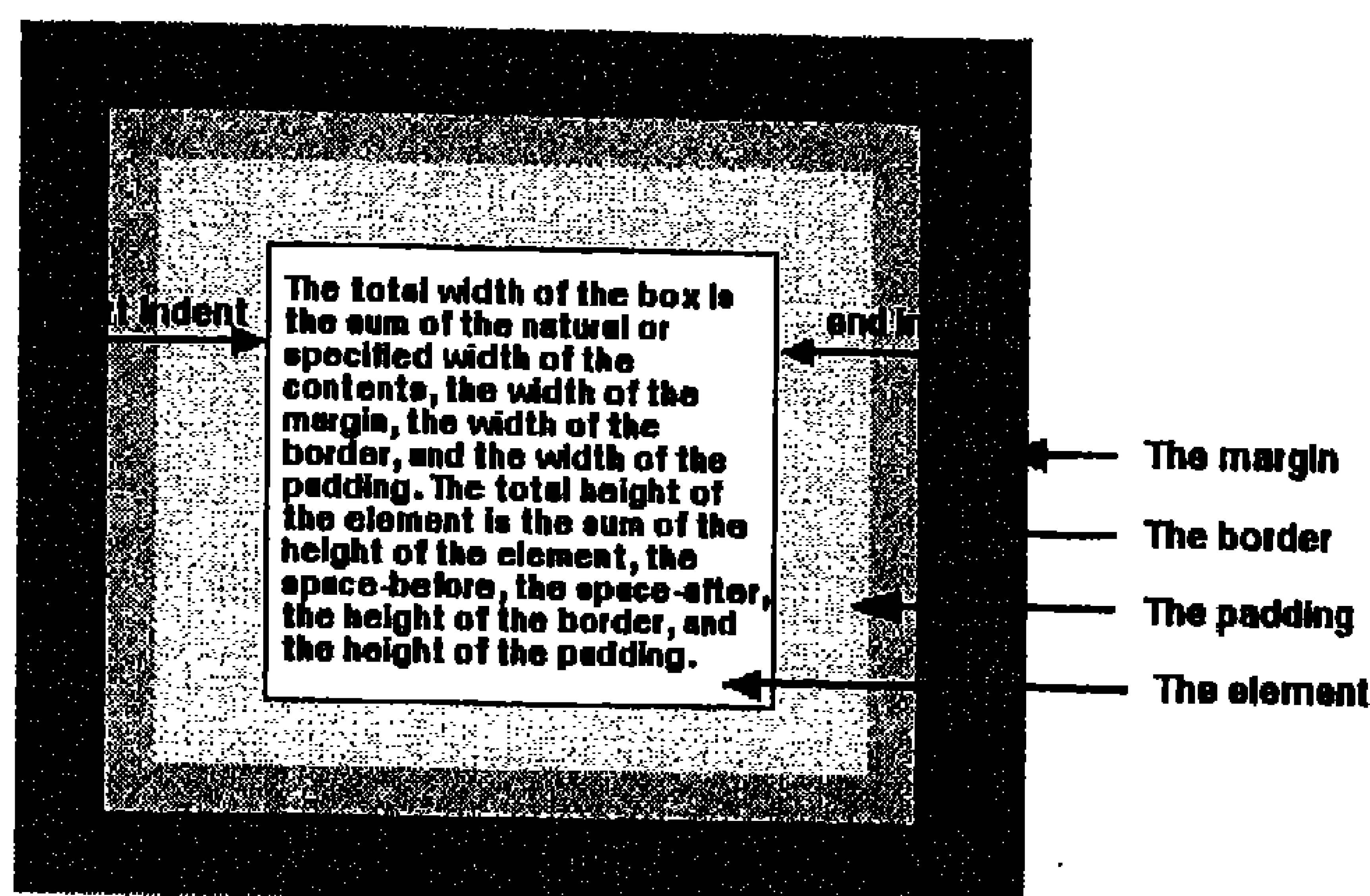
يوجد خمس خصائص هوامش لكل منهما قيمة تعطى كطول غير موقع وهم:

- margin-top
- margin-bottom
- margin-left
- margin-right
- margin

توجد تلك الخصائص فقط لتتوافق مع CSS ولكن من الأفضل استخدام الخصائص التالية والتي تتناسب بصورة أكثر مع نموذج تنسيق XSL:

- space-before
- space-after
- start-indent
- end-indent

تتساوى خصائص space-before وspace-after مع خصائص margin-top و-margin bottom على التوالي وتتساوى خاصية start-indent مع مجموع خصائص padding-left وborder-left-width وmargin-left وتتساوى خاصية padding-right و-border-right width مع مجموع خصائص margin-right يوضح الشكل ١٥-٣ هذا:



الشكل ١٥-٣ Padding ومسافات البادئـــة والحـــدود والمسافات قبل وبعد مربع XSL.

على سبيل المثال، لهذه المجموعة مسافة ٠,٥ سم كهامش في جهتي البداية والنهاية:

```
<fo:block start-indent="0.5cm" end-indent="0.5cm">
  Two strings walk into a bar...
</fo:block>
```

خصائص الهامش للمربعات الضمنية

يطبق على العناصر المحولة نوعين فقط من أنواع خصائص الهامش وهما:

- space-end
- space-start

تعتبر قيمة تلك الخصائص محددات للمسافة التي تعطي مساحة زائدة تتم إضافتها قبل وبعـــد العنصر وقد تكون المسافة الواقعية أقل أو أكبر ولأن المسافة ليست جزء من المربع نفسه فـــإن مسافة نهاية مربع قد تكون جزء من مسافة بدء المربع التالي:

يقدم من محدد المسافة نطاق من محدد المسافة نطاق قيم يتضمن أقصى وأدنى قيمة وأيضـــاً القيمة المثلى. ويعطي المنسق الحرية للاختيار من بين هذه النطاقات ليتلائم مع حدود الصفحـــة.

وبالإضافة إلى ذلك يتضمن محدد المسافة قيم للأسبقية والشرطية ويفصل بين هـؤلاء الخمس بفاصلة منقوطة.

قد تكون الأسبقية إما عدد صحيح أو الكلمة الأساسية force تحدد الأسبقية ما يحدث عندما تتعارض space-end لمساحة محولة مع space-start للمساحة التالية وتفوز المساحة التي لها أسبقية أعلى والأسبقية الافتراضية هي صفر.

أما بالنسبة للشرطية وهي إحدى الكلمتين الأساسيتين conditionality أو retain. تحدد تلك الكلمات الأساسية ما يحدث للمسافة الزائدة في نهاية كل سطر والافتراض هو تجاهلها.

خصائص عرض وارتفاع المحتويات

توجد أربع خصائص تحدد العرض والارتفاع لمساحة المحتويات في المربع كطول غير معروف وتلك هي:

- height
- width
- max-height
- max-width

تلك الخصائص لا تحدد العرض والارتفاع الإجمالي للمربع والتي تتضمن أيضاً الهوامش وpadding والحدود وهذا هو عرض وارتفاع مساحة المحتويات فقط. بالإضافة إلى كونهم طول غير معرف. يمكن إعداد خصائص height وwidth إلى الكلمة الأساسية auto والتي تختار الارتفاع والعرض بناء على كمية المحتوى الموجودة في المربع وعموماً فلا توجد حالة يكون فيها العرض والارتفاع أكبر من القيم المحددة بواسطة خصائص max-height وmax-width. على سبيل المثال:

```
<fo:block height="2in" width="2in">
  Two strings walk into a bar...
</fo:block>
```

خصائص التدفق

تحدد خاصية overflow ما يحدث إذا كان هناك محتوى زائد عن المربع الذي تم تحديد حجمه. قد يكون هذا تحديد واضح باستخدام خصائص الحجم أو تحديد ضمني يعتمد على حجم الصفحة أو أي قيود أخرى هناك أربع احتمالات وكل منها يتم تمثيله بكلمة أساسية:

١- auto: استخدم أشرطة التمرير إذا كان هناك تدفق في المحتوى ولا يستخدموا إذا لم يوجد.

٢- Hidden: لا تظهر أي محتوى يعمل خارج المرجع.

٣- Scroll: أرفق أشرطة التمرير إلى المربع حتى يستطيع القارئ المرور إلى المحتوى الإضافي.

٤- Visible: تعرض المحتويات كاملة، إذا استلزم الأمر، عن طريق التجاوز عن قيود الحجم الموضوعة على المربع.

تحدد خاصية Clip شكل منطقة الالتقاط إذا لم يكن لخاصية overflow قيمة visibleومنطقة الالتقاط الافتراضية هي ببساطة المربع نفسه ويمكن تغيير ذلك عن طريق تحديد مستطيل محدد مثل هذا المربع نفسه ويمكن تغيير ذلك عن طريق تحديد مستطيل محدد قبل هذا:

```
clip=rect(top_offset right_offset bottom_offset left_offset)
```

تعتبر في هذا خصائص top_offset, right_offset, bottom_offset, and left_offset أطوال معرفة تعطي مسافة تعد منطقة سماعة من الجهة العليا اليمنى وأسفل واليسرى للمربع يتيح هذا للمستخدم جعل منطقة clipping أصغر من المربع نفسه.

خاصية توجيه المرجع

تتيح خاصية reference-orientation تحديد أن محتوى المربع يتم تدويره بدرجة ٩٠ زيادة مما يتناسب مع توجيه العادي. القيم الوحيدة الصحيحة المتاحة هي ٩٠ درجة زيادة والتي تم قياسها عكس عقارب الساعة وهذا هو ٩٠، ١٨٠، ٢٧٠ ويمكن كذلك تحديد -٩٠، -١٨٠، -٢٧٠ على سبيل المثال، ما يلي هو استدارة تقدر ٩٠ درجة.

```
<fo:block reference-orientation="90">
  Bottom to Top
</fo:block>
```

خصائص وضع الكتابة

يحدد وضع الكتابة اتجاه النص في المربع ولهذا تأثير مهم بالنسبة لترتيب كائنات التنسيق في المربع. في معظم الحيان، يفترض المستخدمين باللغة الإنجليزية واللغات الغربية الأخرى وضع كتابة من اليسار إلى اليمين ومن أعلى إلى أسفل مثل ما يلي:

A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z

وأما في اللغات العربية والعبرية يبدو الترتيب التالي أكثر طبيعية من اليمين إلى اليسار ومن أعلى إلى أسفل:

G F E D C B A
N M L K J I H
U T S R Q P O
Z Y X W V

في تايوان، يكون الترتيب من أعلى إلى أسفل ومن اليمين إلى اليسار أكثر طبيعية:

A E I M Q U Y
B F J N R V Z
C G K O S W
D H L P T X

في لغة تنسيق XSL. لا يؤثر وضع الكتابة في النص فقط ولكن يؤثر أيضاً في كيفية عرض الكائنات في تدفق أو التالي وكيفية أداء الالتفاف وأكثر من ذلك. لاحظت بالفعل أن العدد من الخصائص قد تم تنظيمها في البداية، النهاية، قبل وبعد المتغيرات بدلاً من يسار، يمين أعلى وأسفل. تحديد قواعد النمط وفقاً للبداية، النهاية، قبل وبعد بدلاً من يسار، يمين أعلى وأسفل ينتج عند أوراق نمط أكثر محلية.

تحدد خاصية writing-mode وضع لكتابة لمساحة. يمكن لهذه الخاصية أن تحصل على واحدة من قيم ١٤ كلمة أساسية:

١- bt-lr: أعلى إلى أسفل، يسار إلى يمين.
٢- bt-rl أسفل إلى أعلى، يمين إلى يسار.
٣- lr-alternating-rl-bt: أسطر يسار إلى يمين بالتبادل مع أسطر يمين إلى يسار أسفل إلى أعلى.
٤- lr-alternating-rl-tb أسطر يسار إلى يمين بالتبادل مع أسطر يمين إلى يسار، أعلى إلى أسفل.
٥- lr-bt: يسار إلى يمين، أسفل إلى أعلى.
٦- lr-inverting-rl-bt: يسار إلى يمين ثم التحرك إلى السطر التالي ثم أذهب يمين إلى يسار وهذا يعني التحرك في الصفحة على هيئة حرف S معكوس.
٧- lr-inverting-rl-tb: يسار إلى يمين، ثم التحرك لأسفل جهة السطر التالي والذهاب إلى اليمين إلى اليسار. التحرك في الصفحة على هيئة حرف معكوس.
٨- lr-tb: يسار إلى يمين، أعلى إلى أسفل.

٩- rl-bt: يمين إلى يسار، أسفل إلى أعلى.

١٠- rl-tb: يمين إلى يسار، أسفل إلى أعلى.

١١- tb-lr: أعلى إلى أسفل، يسار إلى يمين.

١٢- tb-rl: أعلى إلى أسفل، ويمين إلى يسار.

١٣- tb-rl-in-rl-pairs: أعلى إلى أسفل، ويمين إلى يسار.

١٤- use-page-writing-mode: أي ما يكون وضع كتابة الصفحة التي يظهر عليها الكائن يعم استخدام الافتراضي.

اليتامى والأرامل

بالنسبة لمحدد النوع، يعتبر اليتيم هو منفرد لفقرة في نهاية الصفحة أما الأرمل فهو سطر منفرد لفقرة في أعلى الصفحة يتحرك محددات النوع سطر إضافي من الصفحة السابعة أو إلى الصفحة التالية مما يتطلب يتجنب الأرامل واليتامى يمكن تعديل عدد الأسطر المفترض أن تكون يتامى عن طريق إعداد خاصية orphans لعدد صحيح عن طريق إعداد خاصية widows لعدد صحيح غير معرف. فعلى سبيل المثال، إذا أردت التأكد أن كل فقرة جزئية في نهاية الصفحة لها على الأقل أسطر يتم إعداد خاصية orphans على ٣ مثال:

```
<fo:simple-page-master page-master-name="even"
   orphans="3" page-height="8.5in" page-width="11in"
/>
```

Aural Properties

XSL supports the full collection of CSS2 aural stylesheet properties including:

- azimuth
- cue
- cue-after
- cue-before
- elevation
- pause
- pause-after
- pause-before

- ♦ pitch
- ♦ pitch-range
- ♦ play-during
- ♦ richness
- ♦ speak
- ♦ speak-header
- ♦ speak-numeral
- ♦ speak-punctuation
- ♦ speech-rate
- ♦ stress
- ♦ voice-family
- ♦ volume


خصائص ورقة النمط السمعي تم مناقشتها في المقطع الأخير من الفصل ١٣ "أوراق نمط المستوى ٢" ولديهم نفس الدالات وبناء الجملة كائنات تنسيق XSL كما هم في CSS2.

خلاصة

في هذا الفصل تعرفت على لغة تنسيق XSL بالتفصيل وقد تعلمت:

- ♦ تحويل XSL يتم عرضه لتحويل مستند المصدر إلى مستند XML جديد لترميزه في معجم كائن تنسيق XSL.
- ♦ رقم تشغيل كائنات تنسيق أغلب XSL إحدى مساحات المستطيلة وأكبر وتحتوي مساحات الصفحة مساحات للصفحة وتحتوي مساحات الصفحة وتحتوي مساحات الكتل مساحات كتل ومساحات أسطر وتحتوي مساحات السطر على مساحات محولة وتحتوي المساحات المحولة على مساحات محولة أخرى ومساحات أخرى.
- ♦ العنصر الجذري من مستند كائن التنسيق هو fo:root. ويحتوي هذا عناصر fo:layout-master-set وعناصر fo:page-sequence elements.

- تحتوي عناصر fo:layout-master-set واحد أو أكثر من عناصر fo:simple-page-master كل منها يحدد التخطيط من النوع المحدد من صفحة بقسمته إلى ٥ أجزاء وتضمين خصائص لكل واحد.
- تحتوي عناصر fo:page-sequence على عنصر fo:sequence-specifier أو عناصر fo:static-content والعنصر fo:flow وقد تم نسخ fo:flow في أمثلة من الصفحات الرئيسية في ترتيب محدد بواسطة عنصر fo:sequence-specifier ومحتويات من عناصر fo:static-content يتم نسخها على كل صفحة أنشأتها.
- ينتج عنصر fo:display-rule سطر أفقي مستوى الكتلة تنتج fo:inline-rule سطر أفقي محول.
- يجمل عنصر fo:display-graphic صورة من URL وعرضه في كتلة تحميل عنصر fo:inline-graphic صورة من URL وعرضها محولة.
- عنصر fo:simple-link ينشأ رابطة سطر تشعبي إلى URL ويعرض هذا في كتلة.
- قائمة عنصر مستوى الكتلة تم إنشائه بواسطة عنصرfo:list-block وتحتوي مستوى كتلة عناصر fo:list-item يحتوي كل fo:list-item contains على fo:list-item-label وfo:list-item-body.
- تدرج عنصر fo:page-number الرقم الحالي للصفحة.
- يستبدل عنصر fo:character حرف معين أو سلسلة من حروف في مدخلات مع حرف مختلف في المخرج.
- عناصر fo:display-sequence وfo:inline-sequence هي حاويات مستخدمة لربط خصائص إلى النص والمساحات التي تحويها.
- تدرج fo:footnote عنصر مستوى كتلة خارجي ومرجع حاشية يطلب إلى الصفحة.
- يدرج عنصر fo:float عنصر مستوى كتلة مثل شكل أو pullquote في الصفحة
- هناك أكثر من ٢٠٠ خاصية تنسيق XSL منفصلة والكثير منها مطابق لخصائصها CSS من نص الاسم وقد ربطها إلى عناصر كائن تنسيق كسمات

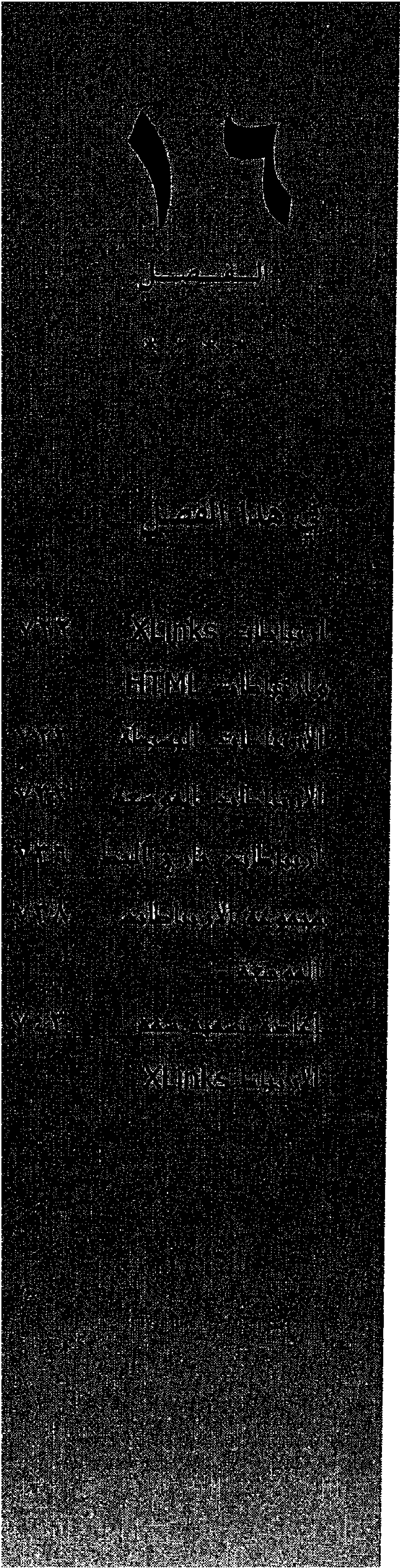
الفصل التالي يعرض Xlinks وهو أكثر بناء جملة قوية ثم HTML الهامش عنصر مرتبط أو fo:display-link وfo:inline الخاص بـ s XSL.

الجزء

٤

في هذا الجزء

تقنيات إضافية


١٦
الفصل
في هذا الفصل
ارتباطات XLinks
مقارنة بـ HTML
الارتباطات البسيطة
الارتباطات الممتدة

XLinks

تنقسم XLL "لغة الربط الممتدة eXtensible Linking Language" إلى جزأيـــن همـــا Xlinks وXPointers. تحدد Xlink المعروفة باسم XML Linking Language كيفية ربط مستند بـآخر. كما تحدد XPointer وXML Pointer Language كيفية العمل مع أجزاء مستقلة من المســـتند. تشير Xlinks إلى URI "وهي URL" التي تحدد مورد معين. وقد يتضمن URL جـزء XPointer الذي يحدد الجزء أو المقطع المطلوب للمورد أو المستند المستهدف. ويتناول هذا الفصل Xlinks، بنما يتناول الفصل التالي Xpointers.

ارتباطات XLinks وارتباطات HTML

يقوم الموقع بإخضاع بروتوكول جوفر الأكثر تثبيتاً لسبب رئيـس واحـد، حيـث يمكـن لـهذه الارتباطات تضمين الصور أو السماح للمستخدم بالانتقال من داخل مستند إلى مستند آخر أو الـى جزء آخر في نفس المستند لدرجة أن لغة XML تختلط بتنسيق آخر مثل HTML في رؤية نفـس بناء الجملة الذي يستخدمه للارتباط بحيث يمكن أن يستخدم في مستندات XML ويمكن تحويل بناء الجملة البديلة إلى بناء جمل HTML باستخدام XSL كما رأيت في أمثلة متعددة في فصل ١٤.

ومع ذلك توجد حدود لارتباطات HTML لأن URL تكون غالباً محددة بالإشارة إلـى مسـتند مفرد والأكثر من ذلك أن الارتباط بالجملة الثالثة في الفقرة ١٧ في المستند الذي يتطلـب منـك إدخال يدوي للاسم المعتمد في الملف المستهدف، ولا يمكن أن تتم بدون كتابـة الوصـول إلـي المستند الذي ترتبط به.

والأكثر من ذلك ألا تقوم ارتباطات HTML بتدعيم العلاقات السابقة بين المستندات ومع ذلـك قد يتبع المستعرضات الطريقة المتبعة خلال سلسلة من المستندات. ومثل هذا التعقـب لا يمكـن الثقة به كثيراً، ولا توجد طريقة في داخل HTML لمعرفة مـن أيـن أتـي القـارئ. وتكـون الارتباطات ذات طريق واحد وتكون مستندات الارتباطات معروفة للمرتبط بها وليس العكس.

وتقدم XLL اقتراح لكثير من الارتباطات الفعالة بين المسـتندات، وقـد صممـت خصيصـاً للاستعمال مع مستندات XML وهناك بعض الأجزاء التي يمكن أن تستخدم مع ملفـات HTML. كما حققت XLL كل ما يمكن تحقيقه، وإلى جانب ذلك تم تدعيم الارتباطات متعددة اللغات عندمـا تعمل الارتباطات في أكثر من اتجاه كما يمكن لأي عنصر أن، يكون له ارتبـاط وليـس فقـط العنصر A، وليس من الواجب أن تخزن الارتباطات في نفس الملف كما في المسـتندات التـي يربطها. والأكثر من ذلك أن يسمح جزء مؤشر X. "وهذا سوف يتم شرحه في الفصـل القـادم" للارتباطات بمواقع معينة في مستند XML وتجعل هذه الميزات XLL مناسباً أكـثر ليـس فقـط للمستخدمين الجدد ولكن للأشياء التي تحتاج لمجهود كبير في HTML مثل المراجع والحواشـي السفلية والتعليقات الختامية وبيانات داخلية مرتبطة وغيرها.



يجب أن يوضع في الاعتبار أنه في وقت كتابة هذه السطور في ربيع ١٩٩٩ كان XLL لا يزال تحت الإنشاء والتطوير والتعديل وعلى الرغم من ذلك قد بدأت في تثبيت بعض الأجزاء التي يمكن أن تتغير عند قراءة هذه السطور.

وبالإضافة إلى ذلك، لا توجد تطبيقات ذات أغراض عامة، والتي تدعم ثوابت ارتباطات X، وذلك لأن ارتباطات XLinks بها الكثير من القواعد العريضة للتطبيق أكثر من ارتباطات HTML ولا تستخدم ارتباطات XLinks فقط مع اتصالات النصوص المترابطة والصور المتضمنة في المستندات. ولكنه يمكن استخدامهم بواسطة أي تطبيق عادي يحتاج إلى إيجاد اتصال بين المستندات وأجزاء المستندات لأي سبب، حتى عندما يكون ارتباط XLinks قد تم تنفيذه في المستعرضات. وقد لا يكون دائماً نص أزرق تحته خط تنقله إلى صفحة أخرى. فقد يكون كذلك أو لا يكون وذلك حسب الحاجة.

الارتباطات البسيطة

يتم تعريف الارتباطات HTML بواسطة علامة <A> ومع ذلك كما أن XLL تكون أكثر مرونة مع العلامات التي تصف العناصر فإنها أكثر مرونة مع العلامات التي تشير إلى العناصر الخارجية، وغالباً يمكن لأي علامة في XML أن يكون ارتباط. تسمى العناصر التي تحتوى على ارتباطات عناصر ربط.

ويتم تعيين عناصر الربط بواسطة سمة XLINK:FORM مع قيمة SIMPLE أو EXTENDED. ومع ذلك يحتوى كل عنصر ربط على سمة HERF والذي تكون قيمته في مصدر URL المرتبط به. وعلى سبيل المثال إليك ثلاثة عناصر ربط.

```
<FOOTNOTE xlink:form="simple"
href="footnote7.xml"<7</FOOTNOTE>
<COMPOSER xlink:form="simple" inline="true"
   href="http://www.users.interport.net/~beand/">
   Beth Anderson
</COMPOSER>
<IMAGE xlink:form="simple" href="logo.gif"/>
```

يلاحظ أن العناصر لها معاني أسماء تصف محتوياتها أكثر من كونها تصف كيفية سلوك العناصر. وتحتوي سمات العلامات على المعلومات التي تدل على أن هذه العناصر هي ارتباطات.

وهذه الأمثلة الثلاث هي ارتباطات XML بسيطة، وهي تشبه ارتباطات HTML القياسية ويتم دعمها بواسطة تطبيقات البرامج قبل الارتباطات الممتدة الأكثر تعقيداً و"الأكثر فاعلية". لذلك سنبدأ بها وسنناقش الارتباطات الممتدة في القسم القادم.

وفي مثال FOOT NOTE بأعلى اسم الارتباط المستهدف هو HERF حيث تكون قيمته فـــي ارتباطه XM1. 7 URL FOOTNOTE ويؤخذ البروتوكول والمصنف والدليـــل الخـــاص بـــهذا المستند في البروتوكول والمصنف والدليل الخاص بالمستند الذي تظهر به هذه الارتباطات.

في مثال COMPOSER بأعلى اسم سمة الارتباط المستهدف هو HERF أن قيمة سمة HERF هو ثبات HTTP://WWW.USERES.PORTNE.NET/URL المثال الثاني بأعلى وهو IMAGE اسم سمة الارتباط المستهدف HERF فقيمة سمة HERF في علاقته مع LOGO.GIF URL مـــرة أخرى يؤخذ البروتوكول والمضيف والدليل الخاص بهذا المستند فـــي الـــبروتوكول والمضيـــف والدليل الخاص بالمستند الذي تظهر به هذه الارتباطات.

وإذا كان لديك مستند DTD فإن هذه السمة يجب أن تعلن كالأخريات. وعلى سبيل المثال فإن إعلان FOOTNOTE وCOMPOSER وعناصر IMAGE قد يكون مثل ذلك.

```
<!ELEMENT FOOTNOTE (#PCDATA)>
<!ATTLIST FOOTNOTE
   xlink:form   CDATA   #FIXED "simple"
   href         CDATA   #REQUIRED
>
<!ELEMENT COMPOSER (#PCDATA)>
<!ATTLIST COMPOSER
   xlink:form   CDATA   #FIXED "simple"
   href         CDATA   #REQUIRED
>
<!ELEMENT IMAGE EMPTY>
<!ATTLIST IMAGE
   xlink:form  CDATA    #FIXED "simple"
   href        CDATA    #REQUIRED
>
```

وبواسطة هذه الإعلانات يكون اسم XLINK:FORM قيمة محدد ومع ذلك لا توجد حاجة إلـــى وجودها في أمثلة العناصر والتي قد تكتبها الآن أكثر دمجاً. ومثال ذلك:

```
<FOOTNOTE href="footnote7.xml"<7</FOOTNOTE>
<COMPOSER href="http://www.users.interport.net/~beand/">
 Beth Anderson
</COMPOSER>
<IMAGE href="logo.gif"/>
```

ولا يفرض تحويل العناصر إلى عناصر ربط أي قيود على السمات الأخرى أو المكونات الخاصة بهذه العناصر وقد تحتوى ارتباطات العناصر على سمة أو أي سمات أخرى والتي دائماً تهدف الى القيم المفروضة على DTD. وعلى سبيل المثال توجد هنا إعلانات بالعنصرIMAGE أكثر واقعية. لاحظ أنه أغلب السمات لا علاقة لها بالارتباطات.

```
<!ELEMENT IMAGE EMPTY>
<!ATTLIST IMAGE
    xlink:form CDATA   #FIXED "simple"
    href       CDATA   #REQUIRED
    ALT        CDATA   #REQUIRED
    HEIGHT     CDATA   #REQUIRED
    WIDTH      CDATA   #REQUIRED
>
```

صفات المورد المحلي

قد يحتوي العنصر الرابط على عناصر content-role وcontent-title التي تقوم بتوفير المزيد من المعلومات مع المزيد. من التوضيح فيما يخص سبب الارتباط الموجود داخل المستند الذي يظهر فيه فعلى سبيل المثال:

```
<AUTHOR href="http://www.macfaq.com/personal.html"
  content-title="author of the page"
  content-role="whom to contact for questions about this page">
    Elliotte Rusty Harold
</AUTHOR>
```

قد تحتوى عناصر الربط على اختيار CONTENT–ROLE وعنصر CONTENT–TITLE والذي يمدنا بمعلومات إضافية وشروح هامة والغرض من الارتباطات الموجودة داخل المستند والذي تظهر بداخلة وعلى سبيل المثال تصف السمات CONTENT-TITLE و-CONTENT ROLE المصدر المحلي وهو محتويات عناصر الارتباطات وهي Elliotte rusty hrold في هذا المثال. وهذه السمات مع ذلك لا تصف المصادر البعيدة الخاصة بهذا المستند http://www.macfug.com/per في هذا المثال. هكذا فإن هذا المثال يقول أن اليوت روس هارلد لديه لقب "مؤلف الصفحة" وليس في الضروري أن يكون لذلك أي علاقة بالمستند الموجود في http://www.macfag.com/personl.html.

وتستخدم سمات content-title عامةً بواسطة قراءة التطبيقات الخاصة بالمصطلح XML ليوضح في المعلومات الإضافية للقارئ ربما عن طريق حالة الحاجز المستعرض أو بواسطة

إضافة منبه عندما يحرك المستخدم الماوس فوق عناصر الربط. ومع ذلك ليس مطلوب في التطبيق إظهار هذه المعلومات إلى المستخدم وقد يفعل ذلك بناءاً على اختيار.

تتضمن سمة content-role الغرض الخاص بارتباطات العناصر في المستند وتشبه سمة content-role عملية التعليمات في ذلك عند القصد في إرسال البيانات إلى قراءة التطبيقات الخاصة بالمصطلح XML. على الرغم من أنها ليس لها غرض حقيقي مثلXML والتطبيقات لديها الحرية في تجاهلها.

```
<!ELEMENT AUTHOR (#PCDATA)>
<!ATTLIST AUTHOR
    xlink:form      CDATA   #FIXED    "simple"
    href            CDATA   #REQUIRED
    content-title   CDATA   #IMPLIED
    content-role    CDATA   #IMPLIED
>
```

وصف المورد البعيد

قد يحتوي عنصر الربط على سمتي role وtitle الخياريتين والتي تصف المصدر البعيد وهو المستند أو أي مصدر آخر يشير إليه الارتباط. وعلى سبيل المثال:

```
<AUTHOR href="http://www.macfaq.com/personal.html"
  title="Elliotte Rusty Harold's personal home page"
  role="further information about the author of this page"
  content-title="author of the page"
  content-role="whom to contact for questions about this page">
   Elliotte Rusty Harold
</AUTHOR>
```

وتصف سمتي role وtitle المصدر البعيد وليس العنصر المحلي ويمثل المصدر البعيد في المثال بأعلى المستند في http://www.mcfaq.com/personl.html وهكذا يقول المثال في أعلى بأن الصفحة في http://www.mcfaq.com/personl.html لديها اسم "الصفحة الرئيسية الشخصية لإليوس روش هارولد" والدور "معلومات إضافية عن كاتب هذه الصفحة". وهي ليست عادية لذلك لا يطلب من أجل title أن يكون مماثلاً لمحتويات العنصر TITLE الخاص بالصفحة التي ترتبط بها.

وقد تستخدم قراءة التطبيق الخاصة بالمصطلح XML هاتين السمتين ليظهر للقارئ أكبر قدر من المعلومات. ومع ذلك لا يطلب في التطبيقات أن تظهر هذه المعلومات إلي المستخدم أو يفعل معها شيئاً.

تتضمن السمة role الغرض من المصدر البعيد "الخاص بالذي ترتبط به" ارتباطات المستند "الخاص بالذي ارتبطت منه". وعلى سبيل المثال، قد تفرق بين الحواشي السفلية والتعليقات الختامية والنص.

يجب أن تعلن السمتان title وrole في DTD لكل العناصر التي تنتمي إليها كما يحدث مع كل السمات الأخرى. وعلى سبيل المثال، هذا إعلان معقول لعنصر المؤلف بأعلى.

```
<!ELEMENT AUTHOR (#PCDATA)>
<!ATTLIST AUTHOR
    xlink:form      CDATA   #FIXED "simple"
    href            CDATA   #REQUIRED
    content-title   CDATA   #IMPLIED
    content-role    CDATA   #IMPLIED
    title           CDATA   #IMPLIED
    role            CDATA   #IMPLIED
>
```

سلوك الارتباط

يمكن احتواء عناصر الربط لثلاث سمات أخرى والتي تقدم للتطبيقات كيفية اقتران المصدر البعيد بالصفحة الحالية.

١- show

٢- actute

٣- behavior.

وتقترح السمة show كيف أن المحتويات يجب أن تكون منظمة عندما يكون الارتباط نشط وعلى سبيل المثال فتح نافذة جديدة لرؤية المحتويات. كما نقترح سمة actuate ما إذا كان يجب على الارتباط أ، تمنع آلياً و/أو إذا كان هناك طلب معين للمستخدم مرغوب. كما يمكن أن تمدنا السمة behvior بمعلومات مفصلة للتطبيق عن كيفية حدوث حجز الارتباط مثل تأجيل الوقت قبل منع الارتباط.

عرض السمات

لدى السمة show ثلاث قيم قانونية هي repice وnew وembed.

عندما يكون الارتباط مع القيمة replace (عامة بالنقد عليها، على الأقل في المستعرض "GU1") فإن الهدف الخاص بالارتباط يحل مكان المستند الحالي في نفس النافذة. ويكون هذا هو السلوك المهمل الخاص بارتباطات HTML.

```
<COMPOSER href="http://www.users.interport.net/~beand/"
        show="replace">
  Beth Anderson
</COMPOSER>
```

تنشيط الارتباط مع القيمة new يفتح نافذة جديدة عندما ينظم المصدر المستهدف. ويكون ذلك شبيهاً بسلوك الارتباطات الخاصة HTML عندما تصبح السمة blank–target.

```
<WEBSITE href="http://www.quackwatch.com/" show="new">
  Check this out, but don't leave our site completely!
</WEBSITE>
```



لا يتوقع القارئ فتح نافذة جديدة بعد النقر على الارتباط. ولكنهم يتوقعون عندما ينقرون الارتباط سوف تحمل في النافذة الحالية إلا إذا طلبوا تحديداً في الارتباط أن تفتح في نافذة جديدة.

وتكون بعض الشركات معتزة بنفسها لأنهم يعتقدون أنه من المستحيل أن أي مستخدم سوف يترك موقعة. وهكذا فهم يساعدون القراء على فتح نوافذ جديدة وفي أغلب الأوقات هذه الخدمة تكون مؤدية إلي الضيق والاضطراب فلا يمكنك تغيير سلوك توقع المستخدمين بدون سبب جيد. والاحتمال الضعيف هو أنه قد يقضي القارئ بعض الثواني في موقعك أو مشاهدة أكثر من صفحة ورؤية بعض الإعلانات وهو ليس بسبب جيد.

تدخل الارتباط النشط على القيمة embed المصدر المستهدف في المستند الموجود. وهذا يعني بالضبط أن التطبيقات معتمدة. ومع ذلك يمكنك أن تتخيل أنها استخدمت فتدعيم جانب العميل أملاً لصفحات الويب وعلى سبيل المثال هذه العناصر تشمل مباشرة عناصر مفردة الخاصة بأعضاء أسرة وتم نسخهم خارج الملفات المتفرقة توماس كرون أندرسون. XML وصموئيل انجلش أندرسون XML.

```
<FAMILY ID="f732">
   <HUSBAND href="ThomasCorwinAnderson.xml"  show="embed"/>
   <WIFE    href="LeAnahDeMintEnglish.xml"   show="embed"/>
   <CHILD   href="JohnJayAnderson.xml"       show="embed"/>
   <CHILD   href="SamuelEnglishAnderson.xml" show="embed"/>
 </FAMILY>
```

والنتيجة بعد حجز الارتباطات وتضمين محتوياتهم في عنصر family هو شئ مثل ذلك.

```
<FAMILY ID="f732">
```

```
<PERSON ID="p1035" SEX="M">
  <NAME>
    <GIVEN>Thomas Corwin</GIVEN>
    <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
    <DATE>24 Aug 1845</DATE>
  </BIRTH>
  <DEATH>
    <PLACE>Mt. Sterling, KY</PLACE>
    <DATE>18 Sep 1889</DATE>
  </DEATH>
</PERSON>
<PERSON ID="p1098" SEX="F">
  <NAME>
    <GIVEN>LeAnah (Lee Anna, Annie) DeMint</GIVEN>
    <SURNAME>English</SURNAME>
  </NAME>
  <BIRTH>
    <PLACE>Louisville, KY</PLACE>
    <DATE>1 Mar 1843</DATE>
  </BIRTH>
  <DEATH>
    <PLACE>acute Bright's disease, 504 E. Broadway</PLACE>
    <DATE>31 Oct 1898</DATE>
  </DEATH>
</PERSON>
<PERSON ID="p1102" SEX="M">
  <NAME>
    <GIVEN>John Jay (Robin Adair )</GIVEN>
    <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
    <PLACE>Sideview</PLACE>
```

```
      <DATE>13 May 1873</DATE>
    </BIRTH>
    <DEATH>
      <DATE>18 Sep 1889 </DATE>
    </DEATH>
  </PERSON>
  <PERSON ID="p37" SEX="M">
    <NAME>
      <GIVEN>Samuel English</GIVEN>
      <SURNAME>Anderson</SURNAME>
    </NAME>
    <BIRTH>
      <PLACE>Sideview</PLACE>
      <DATE>25 Aug 1871</DATE>
    </BIRTH>
    <DEATH>
      <PLACE>Mt. Sterling, KY</PLACE>
      <DATE>10 Nov 1919</DATE>
    </DEATH>
  </PERSON>
</FAMILY>
```

وعلى الرغم من ذلك توجد كل عناصر person ملفات متفرقة ويعامل عنصر famly كـاملاً كما لو كان في ملف واحد.

مثل كل السمات في الملفات الصالحة يجب أن تعلن سمة show في الإعلان <!ATTLIST> الخاص بعناصر الربط في DTD.

```
<!ELEMENT WEBSITE (#PCDATA)>
<!ATTLIST WEBSITE
    xlink:form CDATA    #FIXED "simple"
    href       CDATA    #REQUIRED
    show       (new | replace | embed) "new"
>
```

سمة الإطلاق

تحتوي سمة عناصر الربط actuat على قيمتين ممكنتين هما user و auto وتحدد قيمـة user افتراض أن الارتباط يمنح فقط عندما وإذا طلبه المستخدم. وفي ناحية أخرى إذا تم ضبط سـمه

عنصر الربط actuate على auto ويمتنع الارتباط في أي وقت ويمنع واحد في المصادر المستهدفة الأخرى الخاصة بعنصر الربط ويكون ذلك مفيداً لمجموعة الارتباطات "سوف نناقش ذلك بأسفل".

يجب إعلان سمة actuate في DTD الخاصة بإعلان <!ATTLIST> لعنصر الربط الذي يظهر به مثل كل السمات في المستندات الصالحة وعلى سبيل المثال:

```
<!ELEMENT WEBSITE (#PCDATA)>
<!ATTLIST WEBSITE
    xlink:form CDATA    #FIXED "simple"
    href       CDATA    #REQUIRED
    show       (new | replace | embed) "new"
    actuate    (user | auto) "user"
>
```

سمة السلوك

اعتادت سمة BEHAVIOR على تمرير بيانات ملزمة في تنسيق ملزم إلى قراءة التطبيقات في البيانات ويتوقع أن تستخدم التطبيقات هذه البيانات لعمل تحديد إضافي عن كيفية سلوك الارتباطات. وعلى سبيل المثال: إذا كنت تريد أن تحدد إذا كان ملف الصوت FONFARE يعمل عند قطع الارتباط فإنك تحتاج إلى كتابة ذلك.

```
<COMPOSER xlink:form="simple"
   href="http://www.users.interport.net/~beand/"
   behavior="sound: fanfare.au">
   Beth Anderson
</COMPOSER>
```

اختصار إلى DTD

يجب أن تكون أنواع وأسماء السمات قياسية فإذا كان لديك أكثر من عنصر ربط واحد في المستند فإنه غالبا ما تكون ملائمة لجعل إعلان السمات مرجع معامل الكينونة وبسهولة يتم تكرار ذلك في كل إعلانات عناصر الربط وعلى سبيل المثال:

```
<!ENTITY % link-attributes
  "xlink:form    CDATA   #FIXED 'simple'
   href          CDATA   #REQUIRED
   behavior      CDATA   #IMPLIED
   content-role  CDATA   #IMPLIED
   content-title CDATA   #IMPLIED
```

اختصار إلى DTD

```
    role          CDATA    #IMPLIED
    title         CDATA    #IMPLIED
    show          (new | replace | embed) 'new'
    actuate       (user | auto) 'user'
    behavior      CDATA    #IMPLIED"
  >
  <!ELEMENT COMPOSER (#PCDATA)>
  <!ATTLIST COMPOSER
    %link-attributes;
  >
  <!ELEMENT AUTHOR (#PCDATA)>
  <!ATTLIST AUTHOR
    %link-attributes;
  >
  <!ELEMENT WEBSITE (#PCDATA)>
  <!ATTLIST WEBSITE
    %link-attributes;
  >
```

ولكن هذا يتطلب أن تكون التطبيقات التي تقرأ ملف XML أنت تفهم أن سـمة VEHAVIOR مع قيمة SOUND: FAN FAR AU تعني أن ملف FAN FARMAU يجب أن يعمل عند قطع الارتباط وأغلب أو كل التطبيقات لا تفهم ذلك ولكن قد يستخدموا السمة behavior كمكان ملائم لتخزين المعلومات غير القاسية التي يفهمونها.

يجب أن يتم إعلان السمة behavior في DTD الخاص بعناصر الربط التي تظهر بها وعلى سبيل المثال : العنصر Composer بأعلى يمكن أن يعلن بهذه الطريقة.

```
<!ELEMENT COMPOSER (#PCDATA)>
<!ATTLIST COMPOSER
   xlink:form  CDATA    #FIXED "simple"
   href        CDATA    #REQUIRED
   behavior    CDATA    #IMPLIED
>
```

الارتباطات الموسعة

تسلك الارتباطات البسيطة أكثر أو أقل مثل الارتباطات القياسية التي تعودت عليها في HTML حيث يحتوي كل منها على مصدر واحد محلة ومرجع لمصدر واحد بعيد فالمصدر المحلي هو مكونات عناصر الربط والمصدر البعيد هو الهدف الخاص بالارتباطات.

تذهب الارتباطات الموسعة إلى ما بعد ما يمكن أن تفعله مع الارتباطات الخاصة بلغة HTML حيث أنها تشمر ارتباطات متعددة اللغات بين الكثير من المستندات وارتباطات خارج الخط وتحدد الارتباطات الموسعة بواسطة سمةXlink: form مع extended مثل ذلك:

```
<WEBSITE xlink:form="extended">
```

وتشير الإمكانية الأولى الخاصة بالارتباطات الموسعة إلى أكثر من هدف. وللسماح بذلك تقوم الارتباطات الموسعة بتخزين الأهداف في عنصر الداخل locator الخاص بعناصر الربط أفضل من السمة المفردة href الخاصة بعناصر الربط كما تفعل أي ارتباط بسيط على سبيل المثال:

```
<WEBSITE xlink:form="extended">Cafe au Lait
  <locator href="http://metalab.unc.edu/javafaq/">
   North Carolina
  </locator>
  <locator
   href="http://sunsite.univie.ac.at/jcca/mirrors/javafaq/">
   Austria
  </locator>
  <locator href="http://sunsite.icm.edu.pl/java-corner/faq/">
   Poland
  </locator>
  <locator href="http://sunsite.uakom.sk/javafaq/">
   Slovakia
  </locator>
  <locator href="http://sunsite.cnlab-switch.ch/javafaq/">
   Switzerland
  </locator>
</WEBSITE>
```

كلا من عناصر الربط نفسها website في هذا المثال والداخل الفردي locator على سمات تطبيق على مصادر بعيدة معينة والتي ترتبط بها مثل role وtitle كما في المثال التالي:

```
<WEBSITE xlink:form="extended" content-title="Cafe au Lait"
```

```
       content-role="Java news">
  <locator href="http://metalab.unc.edu/javafaq/"
    title="Cafe au Lait" role=".us"/>
  <locator
    href="http://sunsite.univie.ac.at/jcca/mirrors/javafaq/"
    title="Cafe au Lait" role=".at"/>
  <locator href="http://sunsite.icm.edu.pl/java-corner/faq/"
    title="Cafe au Lait" role=".pl"/>
  <locator href="http://sunsite.uakom.sk/javafaq/"
    title="Cafe au Lait" role=".sk"/>
  <locator href="http://sunsite.cnlab-switch.ch/javafaq/"
    title="Cafe au Lait" role=".ch"/>
</WEBSITE>
```

تنتهي السمات show وbehavior وactuate لو قدمت إلى العناصر الفردية locator.

وفي بعض الحالات كما في المثال بأعلى عندما تشير locators المحددات الفردية إلي النسخ المعكوسة الخاصة بنفس الصفحة فإنه سمات المصدر البعيد الخاصة بالعنصر الفـــردي locator قد تكون متماثلة عبر عناصر الربط ويمكنك استخدام سمات المصدر البعيد فلي عناصر الربـــط نفسها في هذه الحالة وتطبق هذه السمات على كل توابع locator والتي لا تعلن قيم متعارضـــة للسمات المتماثلة. وعلى سبيل المثال

```
<WEBSITE xlink:form="extended" content-title="Cafe au Lait"
   content-role="Java news" title="Cafe au Lait">
   <locator href="http://metalab.unc.edu/javafaq/" role=".us"/>
   <locator
    href="http://sunsite.univie.ac.at/jcca/mirrors/javafaq/"
    role=".at"/>
  <locator href="http://sunsite.icm.edu.pl/java-corner/faq/"
    role=".pl"/>
  <locator href="http://sunsite.uakom.sk/javafaq/" role=".sk"/>
  <locator href="http://sunsite.cnlab-switch.ch/javafaq/"
    role=".ch"/>
</WEBSITE>
```

اختصار آخر إلى DTD

إذا كان لديك الكثير من الارتباطات وعناصر locator فإنه قد يكون من المفيد أن تعرف السمات المتداولة في معامل الكينونة في DTD والتي يمكن إعادة استخدامها في عناصر مختلفة.

```
"role        CDATA          #IMPLIED
  title       CDATA          #IMPLIED
  show        (embed|replace|new) #IMPLIED 'replace'
  actuate     (auto|user)    #IMPLIED 'user'
  behavior    CDATA          #IMPLIED"
>

<!ENTITY % local-resource-semantics.att
  content-title CDATA          #IMPLIED
  content-role  CDATA          #IMPLIED"
>
<!ENTITY % locator.att
  "href        CDATA          #REQUIRED"
>

<!ENTITY % link-semantics.att
  "inline      (true|false)   'true'
  role        CDATA          #IMPLIED"
>

<!ELEMENT WEBSITE (locator*) >
<!ATTLIST WEBSITE
  xlink:form    CDATA    #FIXED "extended"
  %local-resource-semantics.att;
>

<!ELEMENT locator EMPTY>
<!ATTLIST locator
  xlink:form    CDATA    #FIXED "locator"
```

اختصار آخر إلى DTD

```
    %locator.att;
    %link-semantics.att;
  >
```

كما يحدث عادة في المستندات الصالحة يجب أن تعلن عناصر الربط وكل سماتهم الممكنة في DTD. وعلى سبيل المثال في هذا المثال يتم الإعلان عن العناصر المستخدمة في المثال بأعلى مثلهم مثل سماتهم.

```
<!ELEMENT WEBSITE (locator*) >
<!ATTLIST WEBSITE
  xlink:form      CDATA    #FIXED "extended"
  content-title  CDATA    #IMPLIED
  content-role   CDATA    #IMPLIED
  title          CDATA    #IMPLIED
>
<!ELEMENT locator EMPTY>
<!ATTLIST locator
  xlink:form      CDATA    #FIXED "locator"
  href           CDATA    #REQUIRED
  role           CDATA    #IMPLIED
>
```

ارتباطات خارج الخط

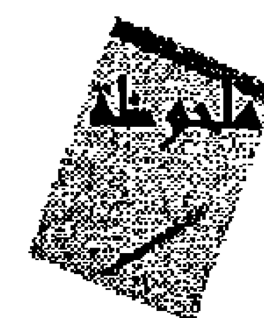
الارتباطات المعنية حتى الآن سواء كانت البسيطة أو الموسعة هي ارتباطات على الخط، مثل عنصر <A> المعروف في HTML والذي يحتوي على الارتباطات. والتي تظهر إلى القارئ. كذلك ارتباطات XLinks ممكن أن تكون خارج الخط فقد لا تظهر الارتباطات التي خارج الخط في أي من المستندات التي تحتويها. وبديل ذلك تخزن الارتباطات في مستندات متفرقة الارتباطات. وعلى سبيل المثال.

ذلك ممكن أن يكون مفيداً في المحافظة على إظهار الشرائح عندما تتطلب كل شريحة الارتباطات السابقة واللاحقة في كل صفحة بدون تحرير الشرائح بواسطة تغيير ترتيب الشرائح في المستند المرتبط.

تمنع السمة in line والقيمة False لتعليم الارتباط بأنه غير محول وعلى سبيل مثال بسيط تصف فيه الارتباطات الغير محولة موقف ويب يستخدم عنصر فارغ. ولا يحتوي العنصر الفارغ على مكونات وفي حالة الارتباط لا يكون لديها مصدر محلي لذلك لا يجب أن تحتوي على

السمات content–role أو content–title والتي تصف المصدر المحلي ولكنها قد تحتوي كما في هذا المثال على السمات role وtitle والتي تصف المصدر البعيد.

```
<WEBSITE xlink:form="simple" inline="false"
      href="http://metalab.unc.edu/xml/"
      title = "Cafe con Leche" role="XML News"/>
```



بسبب أن كل الارتباطات التي مرت علينا حتى الآن هي ارتباطات محولة فهي تتضمن السمات inline مع القيمة true الافتراضيين.

كما في المثال السابق بأعلى الارتباطات غير المحولة البسيطة تكون قليلة نسبياً. ويكون أكثر شويعاً وإفادة الارتباطات الموسعة غير المحولة كما يظهر بأسفل.

```
<WEBSITE xlink:form="extended" inline="false">
  <locator href="http://metalab.unc.edu/javafaq/" role=".us"/>
  <locator
   href="http://sunsite.univie.ac.at/jcca/mirrors/javafaq/"
   role=".at"/>
  <locator href="http://sunsite.icm.edu.pl/java-corner/faq/"
   role=".pl"/>
  <locator href="http://sunsite.uakom.sk/javafaq/" role=".sk"/>
  <locator href="http://sunsite.cnlab-switch.ch/javafaq/"
   role=".ch"/>
</WEBSITE>
```

كما يمكن تخزين مثل هذه الأشياء في ملفات متفرقة الخاصة بخادم الويب في مكان معروف بحيث يستطيع المستعرض إيجادها أو السؤال عنها ليحدد أقرب انعكاس للصفحة التي يبحثون عنها ومع ذلك فإن الحالة غير المحولة هي العنصر الذي لا يظهر في المستند الذي يصبح فيه الارتباط نشطاً.

هذا يوسع فكرة ورقة النمط في مجال الارتباطات حيث تبعد كلياً ورقة النمط عن المستند الذي تصفه وتقوم الآن بمنح قوانين توضح كيفية تقديم المستند إلى القارئ. كما يبعد المستند المرتبط المحتوي على ارتباطات غير محولة عن المستند المتصل بها حيث تقوم الآن بتقديم الارتباطات اللازمة إلى القارئ وهي تحتوي على مزايا متعددة والتي تشمل الاحتفاظ بعرض مبني أكثر على التركيز مبتعداً عن المستندات ويسمح بالارتباطات الخاصة بالمستندات التي تقوأ فقط.



تكون ورقة النمط أكثر حلولاً في الارتباطات غير المحولة. كما لم يقترح تداولهم عامة كيفية ربط ورقة النمط بمستندات XML أو على الأقل كيفية تحديد أي العناصر المفردة الخاصة بالمستند وتكون ملحقة بأي الارتباطات.

هناك اختيار واضح وهو إضافة كعملية التعليمات إلى التسجيل المبدئي الخاص بالمستند لتحديد أين يوجد الارتباط. كما تستطيع ارتباطات الورقة نفسها استخدام شئ مماثل لأنواع XLS المختارة لينظم الارتباطات الخاصة بعناصر XML المفردة. كما يمكن أن يصبح المحددون قيم للعنصر LOCATOR الخاص بالسمة ROLE.

مجموعة الارتباطات الموسعة

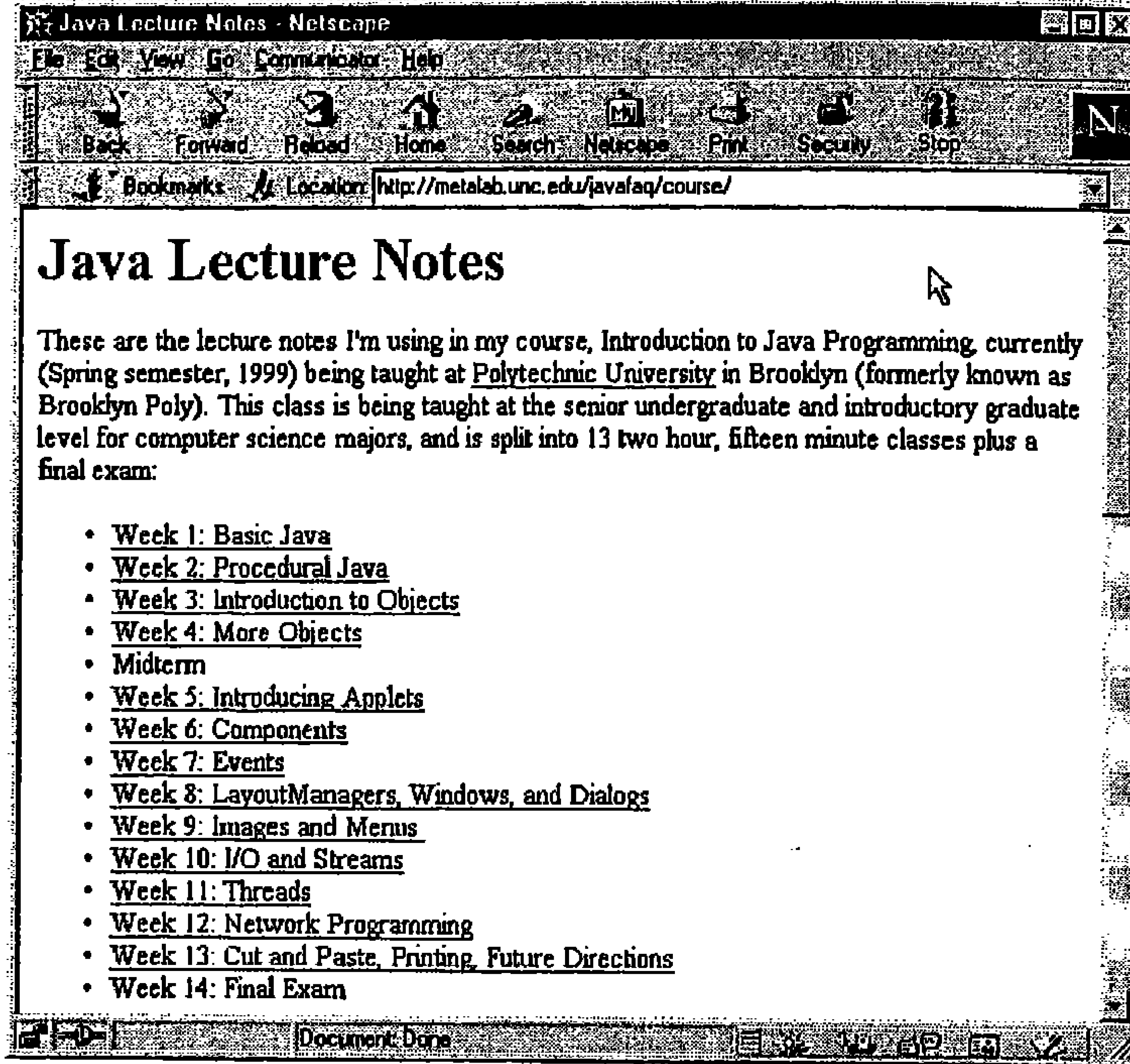
يحتوي عنصر مجموعة الارتباطات الموسعة على قائمة في الارتباطات التي تتصل بمجموعة معينة في المستندات ويتم استهداف كل مستند في المجموعة بواسطة مستند عنصر الارتباطات الموسعة وتكون مسئولية التطبيقات هي فهم كيفية التنشيط وفهم الاتصالات بين أعضاء المجموعة.



أشعر بأنه من المفروض أن أذكر الملحوظة الخاصة بتدعيم التطبيقات الخاصة بارتباطات المجموعات في أحسن افتراضاتها في أثناء وقت هذه الكتابة مع ذلك يمكننا عرض كيفية كتابة مثل هذه الارتباطات وأدواتهم وتدعيماتهم ومن المحتمل فيما بعد وتبقى بعض التفصيلات لتعرف ومن المحتمل أن تنفذ طريقة السوق المحددة في النهاية مبدئياً، وماذا لو يحملون الوعد بإمكانية تمكين الارتباطات الأكثر صقلاً في التي يمكن تحقيقها مع HTML.

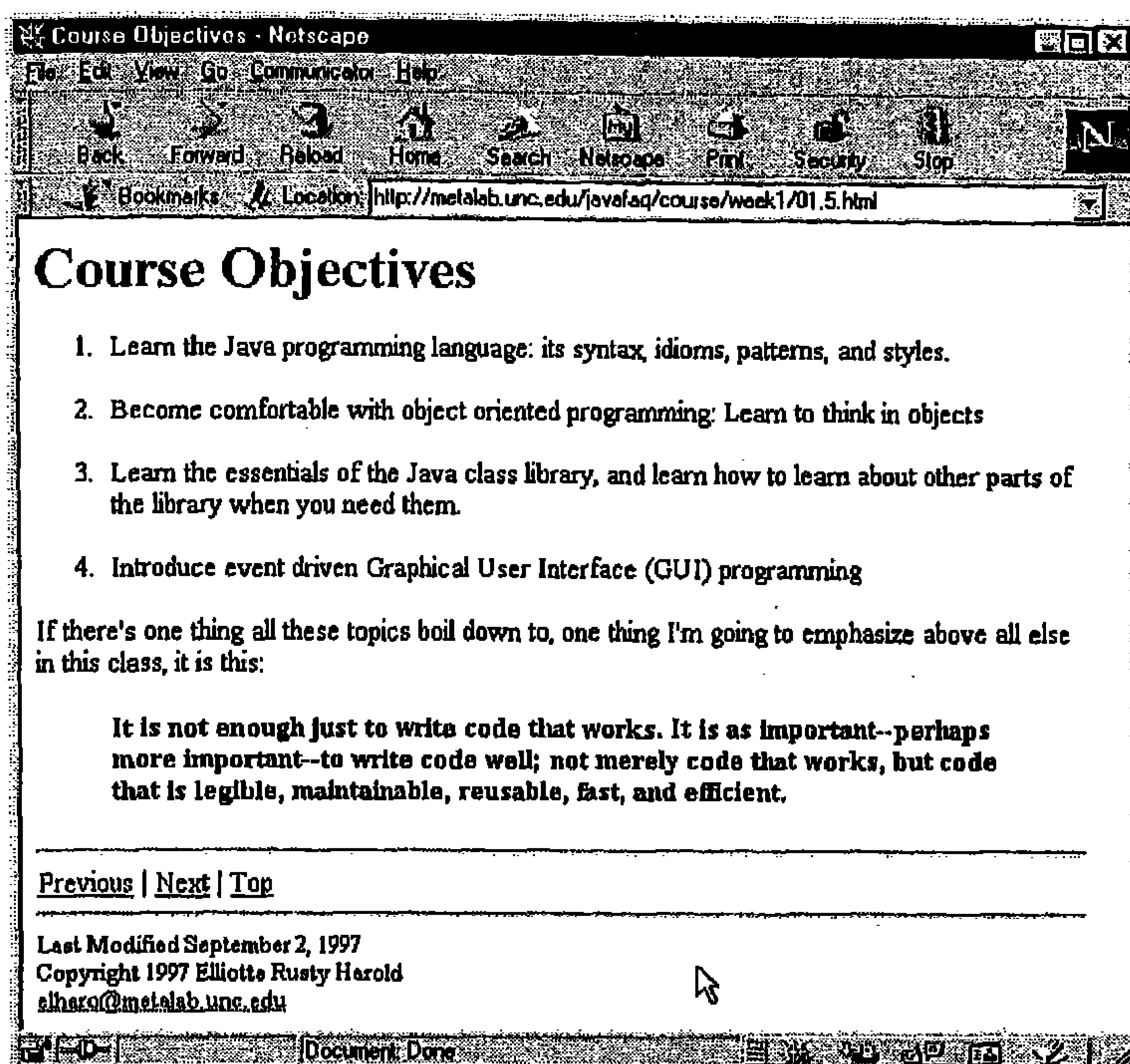
مثال

وعلى سبيل المثال لقد وضعت الملاحظات الخاصة بمحاضرات JAVA والتي درستها موقع ويب الخاص بي، يظهر الشكل ١-١٦ صفحة المقدمة. وتتكون هذه المحاضرات من ١٣ فصل كل منهم يحتوي ما بين ٣٠ و٦٠ صفحة مفردة من الملاحظات. ويمنح بعد ذلك جدول بالمحتويات لكل فصل، تقوم كل صفحة من المائة صفحة المتنوعة بترتيب الموقع الداخلي الذي يحتوى على ارتباطات خاصة بالمستندات السابقة والمستندات التالية وجدول المحتويات الخاص بالأسبوع "الارتباطات العليا" كما يوضح في الشكل ٢-١٦ بوضع الجميع على بعض تصبح الكمية اكثر من ألف اتصال داخلي داخل هذه المجموعة في المستندات.



الشكل ١٦-١ تظهر صفحة المقدمة الخاصة بفصل موقع الويب ١٣ أسبوع من ملاحظات المحاضرات.

تنمو الاتصالات الداخلية الممكنة على نحو مفصل على عدد المستندات. حيث يتم في كل وقت إزاحة مستند أو إعادة تسميته أو تقسيمه إلى أجزاء أصغر فالارتباطات يجب أن تنظم في هذه الصفحة والصفحات التي قبلها وبعدها في المجموعة وفي جدول المحتويات الخاص بالأسبوع وبصراحة فهي تعمل أكثر مما يجب أن تكون عليه وهي تميل إلى التقليل من التشجيع على القيام بالتعديلات الضرورية ومواكبة ملاحظات المحاضرات.



الشكل ١٦-٢ تنظم صفحة واحدة من ملاحظات المحاضرات الارتباطات السابقة والتالية والعليا.

الشيء الوحيد الذي يمكن عمله إذا كانت HTML تدعمه هو تخزين الاتصالات في مستندات متفرقة، كما تتم عمليه إعادة تنظيم الصفحات بواسطة إضافة مستند واحد، حيث لا تدعم ارتباطات HTML ذلك ولكن ارتباطات XLinks تدعمه. وبدلاً من تخزين الارتباطات المحولة في ملفات HTML ممكن أن يخزنوا بدون تحويل في مجموعات العناصر. وعلى سبيل المثال:

```
<COURSE xlink:form="group">
  <CLASS xlink:form="document" href="week1/index.xml"/>
  <CLASS xlink:form="document" href="week2/index.xml"/>
  <CLASS xlink:form="document" href="week3/index.xml"/>
  <CLASS xlink:form="document" href="week4/index.xml"/>
  <CLASS xlink:form="document" href="week5/index.xml"/>
  <CLASS xlink:form="document" href="week6/index.xml"/>
  <CLASS xlink:form="document" href="week7/index.xml"/>
  <CLASS xlink:form="document" href="week8/index.xml"/>
  <CLASS xlink:form="document" href="week9/index.xml"/>
```

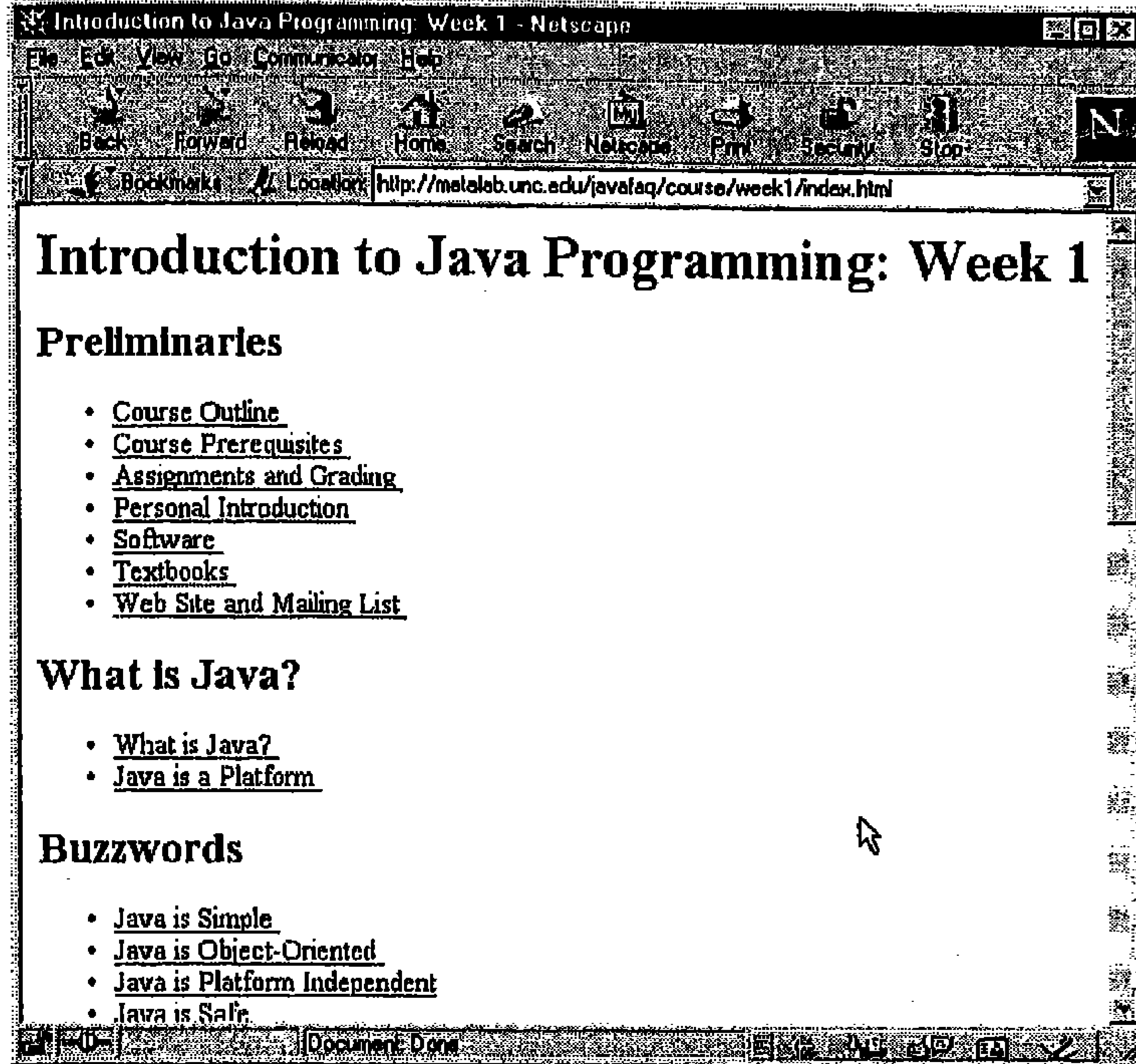
```
  <CLASS xlink:form='document" href="week10/index.xml"/>
  <CLASS xlink:form="document" href="week11/index.xml"/>
  <CLASS xlink:form="document" href="week12/index.xml"/>
  <CLASS xlink:form="document" href="week13/index.xml"/>
</COURSE>
```

هذا يعرف العنصر COURSE على أنه مجموعة من الارتباطات الموسعة والتي تحتوي على ١٣ مستند من عناصر الارتباطات الموسعة ألا وهي العنصر CLASS

سمة الخطوات

قد يختار التطبيق شئ واحد ليفعله مع مجموعة الارتباطات وهي بداية تحميل كل المستندات في مجموعة الارتباطات، وقد تحتوي هذه المستندات على مجموعة الارتباطات الخاصة بهم. وعلى سبيل المثال : كل من عناصر CLASS بأعلى يشير إلى واحد من جداول المواقع الخاص بمحتويات الصفحات الخاصة بأسبوع معين كما يظهر في الشكل ٣-١٦، وبذلك يمكن لهذه المستندات أن تحمل، وعلى سبيل المثال فإن الملف WEEK/INDEX .XML يمكن أن يحوى مجموعة الارتباطات هذه:

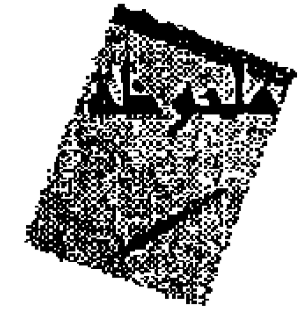
```
<CLASS xlink:form="group">
  <SLIDE xlink:form="document" href="01.xml"/>
  <SLIDE xlink:form="document" href="02.html"/>
  <SLIDE xlink:form="document" href="06.html"/>
  <SLIDE xlink:form="document" href="12.html"/>
  <SLIDE xlink:form="document" href="13.html"/>
  <SLIDE xlink:form="document" href="16.html"/>
  <SLIDE xlink:form="document" href="17.html"/>
  <SLIDE xlink:form="document" href="19.html"/>
  <SLIDE xlink:form="document" href="21.html"/>
  <SLIDE xlink:form="document" href="22.html"/>
  <SLIDE xlink:form="document" href="24.html"/>
</CLASS >
```



الشكل ١٦-٣ تظـــهر صفحـــة جـــدول المحتويـــات ملاحظـــات محاضرات الأسبوع الأول.

يشير أحد هذه المستندات المقترحة إلى الرجوع إلى المستند الأصلي وقد يؤدي ذلك إلى بدايـة استرجاع لا نهائي مع نفس المستندات التي تعيد التحميل حتى تنتهي التطبيقات الخاصة بـالذاكرة، ولمنع ذلك قد تحتوي عناصر المجموعة على السمة steps والتي تحدد عدد المستويات حتى تتبع مجموعة الارتباطات بطريقة recursively. تكرار تركيبي:

```
<group xlink:form="group" steps="3">
```



ولكي أكون أميناً فإني لا أعرف مدى أهمية ذلك، فليس من الصعب على التطبيق أن يكتب ملاحظاته عن متى تعمل التطبيقات فعلاً مع المستند وحتى لا تعمل معـه مرة أخرى. حيث أنني أشك في أنه من السليم وضـــع الطلـــب الخـــاص بمنـــع recursion مع معالج XML أكثر من مؤلف الصفحة. تكرار مركب

يمكن أن تستخدم السمة steps التي تحدد الكمية الخاصة بالتحميل الأولى الموجـــود. فعلـــى سبيل المثال، في مثال ملاحظات الفصل حيث أنه ليس من المحتمل أن الشـــخص ســـوف يقـــوم

بقراءة المجموعة الداخلية الخاصة بملاحظات المحاضرات في جلسة واحدة لذلك قد يحتاج هو أو هي إلى طباعتها أو نسخها جميعها، وعلى أيه حال بوضع السمة steps إلى العدد ١ يمكنك تحديد عمق الانتقال بتسهيل اسم الصفحات أفضل من المائة صفحة المتنوعة في المحاضرات.

وكالعادة يجب أن تعلن هذه العناصر والسمات الخاصة بها في DTD الخاصة بأي مستند صالح والذي يظهروهن به، وعملياً يتم تعيين السمة xlink:form لذلك فهي لا تحتاج أن يتم شمولها في الأمثلة الخاصة بالعناصر وعلى سبيل المثال :

```
<!ELEMENT CLASS (document*)>
<!ATTLIST CLASS
   xlink:form  CDATA   #FIXED   "group"
   steps       CDATA   #IMPLIED
  >
<!ELEMENT SLIDE EMPTY>
<!ATTLIST SLIDE
   xlink:form   CDATA  #FIXED   "document"
   href         CDATA  #REQUIRED
>
```

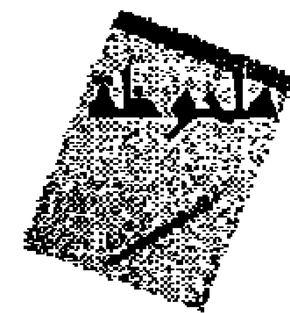
إعادة تسمية سمة الارتباط XLinks

يتم بناء الارتباطات XLinks حول السمات العشرة التي تمت مناقشتها في القسم السابق والموجودة بأسفل.

xlink:form
href
steps
title
role
content-title
content-role
show
actuate
behavior

لا يمكن أن نتصور أن واحدة أو أكثر من هذه السمات أن تستخدم كاسم سمة في تطبيق معين للغة XML ويبدو أن السمة title من المحتمل خصيصاً أن يتم استخدامها بينما السمة الوحيدة التي لا يجب أن تستخدم في غرض آخر هي xlink:form.

يعجل التحديد الخاص بارتباط XLinks بهذه المشكلة ويمكنك في إعادة تسمية سمات ارتباطات XLinks إلى آخر أكثر ملائمة مستخدماً السمة xml:attributes ويتم إعلان هذه السمة كإعلان <ATTLIST> الخاص بلغة DTD كسمة معينة مع طباعة CDATA والقيمة الخاصة بالفراغ الأبيض المتباعد في قائمة خاصة بباريس والخاصة بالأسماء القياسية والأسماء الجديدة.



لقد تم تصميم namespaces "التي تم مناقشتها في الفصل ١٨" لحل هذه المشكلة. ولن أكون مندهشاً عند رؤية هذه التقنية قد تم حذفها من مسودة المستقبل الخاصة بلغة XLL واستبدالها بسوابق namespaces البسيطة مثل :xlink.

وعلى سبيل المثال تبدو عناصر الارتباط الموجودة في هذا الفصل مضحكة لأن الأسماء القياسية تكون جميعها حروف استهلالية صغيرة بينما تكون عادة هذه الكتب جميعها حروف استهلالية كبيرة، لذلك فإنه من السهل أن يتم تغيير سمات الارتباطات XLinks لكي تصبح حروف استهلالية كبيرة مع الإعلان مثل ذلك.

```
<!ELEMENT WEBSITE (#PCDATA)>
<!ATTLIST WEBSITE
  xlink:form CDATA    #FIXED "simple"
  xml:attributes CDATA #FIXED
      "href HREF show SHOW actuate ACTUATE"
  HREF      CDATA    #REQUIRED
  SHOW       CDATA    (new | replace | embed) "new"
  ACTUATE    CDATA    (user | auto) user
>
```

يمكنك الآن إعادة كتابة مثال WEBSITE في هذا الشكل الأكثر انسجاماً:

```
<WEBSITE HREF="http://www.microsoft.com/" SHOW="new">
 Check this out, but don't leave our site completely!
</WEBSITE>
```

يقوم الإعلان ATTLIST أعلاه فقط بتغير السمات الخاصة بالعنصر WEBSITE إذا كانت تريد أن تيتم تغير هم بنفس الطريقة في الأمثلة الأخرى المتعددة فإن أسهل وسيلة لذلك هي استخدام معامل الكينونة.

```
<!ENTITY LINK_ATTS
  'xlink:form    CDATA    #FIXED "simple"
  xml:attributes CDATA #FIXED
       "href HREF show SHOW actuate ACTUATE"
  HREF       CDATA    #REQUIRED
  SHOW       CDATA    (new | replace | embed) "new"
  ACTUATE    CDATA    (user | auto) "user" '
>

<!ELEMENT WEBSITE (#PCDATA)>
<!ATTLIST WEBSITE %LINK_ATTS;>

<!ELEMENT COMPOSER (#PCDATA)>
<!ATTLIST COMPOSER %LINK_ATTS;>

<!ELEMENT FOOTNOTE (#PCDATA)>
<!ATTLIST FOOTNOTE %LINK_ATTS;>
```

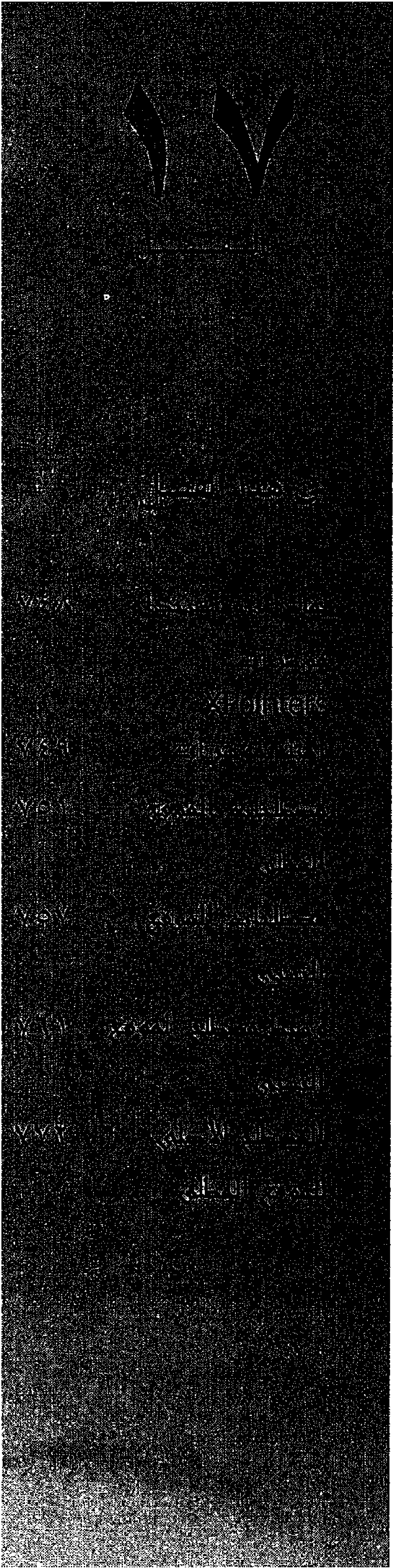
خلاصة

لقد تعرفت في هذا الفصل على ارتباطات XLinks وعلى وجه الخصوص:

- يمكن أن تقوم بارتباطات XLinks القيام بكل ما تستطيع ارتباطات HTML القيام به وأكثر، ولكن لا يتم تدعيمهم بواسطة التطبيقات الحالية.
- يشبه سلوك الارتباطات البسيطة كثيراً السلوك الخاص بارتباطات HTML. ولكنها غير مقصورة على علامة <A>. الفردية
- يتم تعريف عناصر الارتباط بواسطة xlink:form والسمات href.
- يمكن لعناصر الارتباطات ان تصف المصدر المحلي والسمات content-title and content-role.
- يمكن لعناصر الارتباطات أن تصف المصدر البعيد الذي يتم ارتباطهم به مع السمات title وrole.
- يمكن لعناصر الارتباطات أن تستخدم السمة show حتى يتم أخبار التطبيق عن كيفية أن تكون المحتويات منظمة عندما يتم تنشيط الارتباطات أو على سبيل المثال، بواسطة فتح نافذة جديدة.

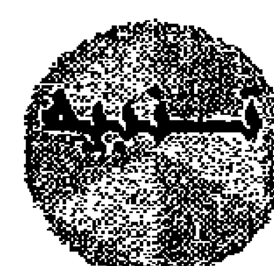
- يمكن لعناصر الارتباطات استخدام السمة behavior حتى يتم إعداد التطبيق بـــالتفصيل، ومعلومات التطبيق المعتمدة عن كيف يتم قطع الارتباط بالضبط.
- يمكن لعناصر الارتباط أن تستخدم السمة actuate لإخبار التطبيق إذا كان يجب أن يتـم قطع الارتباط بدون طلب مستخدم معين.
- يمكن للارتباطات الموسعة أن تشتمل على أكثر من URL فردي في عنصر الارتباطـــات : حالياً، لقد تم ترك ذلك للتطبيق حتى يتم تحديد كيف يتم الاختيار بين البدائل المختلفة.
- تحتوي عناصر مجموعة الارتباطات الموسعة على القائمة الخاصة بالارتباطات والتــــي تتصل بمجموعة معينة من المستندات.
- يمكن أن يتم استخدام السمة xml:attributes الخاصة بلغة DTD لإعادة تســمية ســمة الارتباطات XLink القياسية مثل href وtitle.

سوف نرى في الفصل القادم كيف يمكن أن يتم استخدام مؤشـــرات XPointers ليـــس فقـــط للارتباط مع المستندات البعيدة ولكن مع عناصر أكثر تعيناً في المستندات البعيدة.



مؤشرات XPointers

تقوم مؤشرات XML الخاصة بلغة مؤشرات XML بتحديد نظام العنونة للأجزاء الفردية الخاصة بمستندات XML. كما تقوم ارتباطات XML بالإشارة إلى URL (بين قوسين، URL) والتي تصف المصدر المعين. كما قد يشتمل URL جزء في مؤشرات XML والذي يكون أكثر تخصيصاً في تحديد العنصر أو الجزء المرغوب الخاص بالمصدر المستهدف أو المستند. حيث يقوم هذا الفصل بمناقشة مؤشرات XML.



يقوم هذا الفصل على العمل التمهيدي الخاص بمواصفات مؤشرات XML في ٣ مارس ١٩٩٨. ومن المحتمل أن تكون الصورة الخارجية التي تم تقديمها صحيحة ولكن التفاصيل قد تكون هدف للتقييم. ويمكنك أن تجد العمل التمهيدي الأميز في http:// www.w3.org/TR/WD-xptr

لماذا يتم استخدام مؤشرات XPointers

إن استخدام URL يتم بسهولة وبساطة ولكنه يكون محدود أيضاً وذلك بسبب أن URL تشير فقط إلى مستند عامل فردي. والأكثر تصفية في ذلك، مثل الارتباط بالجملة الثالثة في الفقرة رقم ١٧ في المستند ويتطلب ذلك من المؤلف الخاص بالمستند المستهدف بأن يدخل يدوياً الاسم المعتمد الخاص بالمكان المستهدف. بينما لا يستطيع المؤلف الخاص بالمستند الذي يقوم بالارتباط أن يفعل ذلك، إلا إذا كتب هو أو هي الموصلات إلى المستند الذي يتم الارتباط به، حتى إذا كان المؤلف الذي يقوم بالارتباط قادر على إدخال الأسماء المعتمدة في المستند المستهدف، فإنها تكون غالباً ملائمة.

وقد يكون من المفيد أن إمكانية الارتباط بعنصر أو مجموعة عناصر معينة في الصفحة بدون أن يتم تغيير في المستند الذي ترتبط به. وعلى سبيل المثال، عندما يتم إعطاء صفحة واسعة مثل إحصائيات بيسبول الكاملة الخاصة بالفصل ٤ ، ٥ فقد تريد الارتباط بفريق واحد أو لا عن واحد، حيث توجد أجزاء مختلفة خاصة بهذه المشكلة، وأول هذه الأجزاء هو عنونة العناصر المفردة، وهذا هو الجزء الذي يتم حله بواسطة مؤشرات XML، كما تسمح مؤشرات XML بأن يتم استهداف العنصر المعطى بواسطة الرقم، الاسم، النوع، أو علاقته بالعناصر الأخرى في المستند.

ويكون البرتوكول الجزء الثاني في المشكلة حيث يقوم بواسطة المستعرض بسؤال خادم الويب لإرسال جزء في المستند بدلاً من إرسال الكل، لذلك فهي منطقة بحث وتأمل نشطة ، وتحتاج إلى عمل كثير، حيث تقوم مؤشرات XML بعمل قليل لحل هذه المشكلة فيما عدا إمداد المؤسسات التي يتم فيها بناء مثل هذه الأنظمة. وعلى سبيل المثال، فإن أفضل الجهود لوضع التاريخ هي "اتساع مدى البايت الخاصة بـ HTTP" المزعومة والمتاحة في HTTP 1.1، ومن الآن لم يتم تحقيق تبني واسع الانتشار وفي الأغلب بسبب أن مؤلف الويب لا يكون مرتاحاً عند

وصف طبقة ١ بايت في المستند، وعلاوة على ذلك تكون طبقة بـــايت فـــي غايـــة الهشاشـــة، التحريرات العادية الخاصة بالمستند وحتى إعادة التنسيق البسيط يمكنها تدمير ارتباطـــات طبقـــة بايت، كما تسمح HTTP 1.1 لوحدات طبقات أخرى بجانب البايت الخام "كما في مثال عنـــاصر XML" ولكنها لا تحتاج في خادم الويب أو المستعرض أن يدعموا هذه الوحدات، ويتبقـــى عمـــل كبير ليتم إنجازه.

ويكون الجزء الثالث في المشكلة هو التأكد من أن المستند المسترجع يكون له معنى بدون بقية المستندات ويفي ذلك في محيط XML أن جزء الارتباط تم تكوينه جيداً أو ربما صالحاً. ويكـــون هذا الاقتراح مخادع لأن أغلب مستندات XML وخاصةً المستندات التي مع التسجيل المبدئي الغير مهمل، والتي لا تتحلل جيداً، ومرة أخرى فأن مؤشرات XML لا تعنون ذلك. يقـوم W3C XML جزء مجموعة العمل بعنونة هذا الموضوع، ولكن العمل بدأ الآن فقط.

ومع ذلك، يمكن لمؤشرات XML أن تستخدم في مستند كامل، حيث يتم تحميل الكل وبعد ذلك يتم وضعه في المكان الذي يحدد بواسطة مؤشرات XML. أن المصطلح الطويل والـــذي يمثـــل اتساع لكل من XML وارتباطات XML و HTTP وبروتوكولات أخرى قد تسمح باستخدام رفيـــع ثقافة أكثر الخاص بمؤشرات XML. على سبيل المثال، قد تكون قادراً على اقتباس المستند البعيـد بواسطة الاستعمال فقط على مؤشر XML في الفقرة التي تريد اقتباسها، فضلاً عن إعـــادة نســـخ الموضوع الخاص بالاقتباس كما يمكن أن يتم استعمال المرجع داخل المستند الـــذي يقـــوم آليـــاً بتحديث نفسه كما في المستند الذي تمت مراجعته. يجب على هذه الاستخدامات أن تنتظر التطور الخاص بتكنولوجيا الأجيال القادمة المختلفة، أما الآن، يجب أن تشعر بالسعادة لتحديـــد الجـــزء الخاص بالمستند الذي يجب أن تنتقل إليه عندما يتم اتباع مؤشرات XML على وجه التحديد.

أمثلة مؤشرات

تقوم ارتباطات HTML عامةً بالإشارة إلى مستند معين ومن المستحب أن تكون الإشارة إلى قسم معين وفصل أو فقرة خاصة بمستند معين لا تكون جيدة التدعيم. عندما يتم الإمداد للتحكم في كلاً من الارتباطات ومستندات الارتباطات، حيث يمكنك إدخال الاسم المعتمد داخل الملف HTML في المكان الذي تريد أن ترتبط به، على سبيل المثال:

```
<H2><A NAME="xpointers">XPointers</A></H2>
```

ويمكنك بعد ذلك الارتباط بهذا الموقع في الملف بواسطة إضافة # والاسم الخاص بـــالمعتمد داخل الارتباط، وعلى سبيل المثال، في جدول المحتويات الذي تشاهده:

```
<A HREF="#xpointers">XPointers</A>
```

وعملياً، يكون هذا الحل غير مرغوب فيه، ولا يكون من الممكـــن عـــادةً تعديـــل المســـتند المستهدف حتى يستطيع المصدر الارتباط به، وربما يكون المستند المستهدف موجود على خـــادم

مختلف يتم التحكم فيه بواسطة شخص آخر غير المؤلف الخاص بمستند المصدر وقـــد يغيرهـــا المؤلف الخاص بالمستند المستهدف أو يحذفها بدون ملاحظة المؤلف الخاص بالمصدر.

والأكثر من ذلك يقوم الاسم المعتمد بالتعدي على التباعد الخاص بالترميز المحتوي، ووضـــع الاسم المعتمد في المستند لا يخبر شيئاً عن المستندات أو مكوناتها وتكون فقط علامـــــة خاصــــة بالمستندات الأخرى التي تشير إليها، ولا تضيف شيئاً إلى المحتوى الخاص بالمستندات.

تسمح مؤشرات XML باتصالات أكثر تطوراً بين المستندات في خلال اســـتخدام مؤشـــرات XML كما يمكن أن يشير مؤشر XML إلى عنصر معين خاص بالمستند إلى الأول والثاني أو رقم ١٧ مثل العنصر وإلى العنصر الأول الذي يمثل التابع الخاص بالعنصر المعطى. ويستمر كذلك، وتقوم مؤشرات XML بتقديم اتصالات أكثر قوة بين المستندات، فلا تتطلب أن يحتـــوي المســـتند المستهدف ترميز أكثر لذلك فقط يمكن للقطع المفردة أن ترتبط بها.

والأكثر من ذلك، بخلاف المعتمدات HTML ومؤشرات XML لا يجب الإشارة فقط إلى نقطـة مفردة في المستند. كما يمكن أن يشيروا إلى الطبقات والاجتيازات، وهكذا يمكن أن يتم نسخها أو تحميلها داخل البرنامج.

وتوجد هنا بعض الأمثلة الخاصة بمؤشرات XML:

```
root()
id(dt-xmldecl)
descendant(2,termref)
following(,termdef,term,CDATA Section)
html(recent)
id(NT-extSubsetDecl)
```

ويقوم كلاً منها باختيار جزء معين من المستند، ولا يكون المستند موصفاً في المؤشـــر XML بينما تصف مؤشرات XML المستند، XML تحتوي مؤشرات XML التي شاهدتها فــــي الفصـــل السابق على مؤشرات XML، ولنه ليس من الصعب أن يتم إضافة مؤشرات XML إليها، معظـــم الوقت الذي تضيق فيه ببساطة مؤشرات XML إلى URL التي يتم تباعدها بواسطة # فقـــط كمـــا تفعل مع الأسماء المعتمدة في HTML، وعلى سبيل المثال، يمكن لهذه القائمة أعـــلاه والخاصــــة بمؤشرات XML يمكن أن تضاف إلى URL وتصبح كالآتي:

```
http://www.w3.org/TR/1998/REC-xml-19980210.xml#root()
http://www.w3.org/TR/1998/REC-xml-19980210.xml#id(dt-xmldecl)
http://www.w3.org/TR/1998/REC-xml-
19980210.xml#descendant(2,termref)
```

```
http://www.w3.org/TR/1998/REC-xml-19980210.xml#following(,termdef,term,CDATA Section)
http://www.w3.org/TR/1998/REC-xml-19980210.xml#id(NT-extSubsetDecl)
```

ومن الطبيعي أن يتم استخدام ذلك مثل القيمة الخاصة بالسمة href الخاصة بالعنصر locator، وعلى سبيل المثال:

```
<locator
  href="http://www.w3.org/TR/1998/REC-xml-19980210.xml#root()">
  Extensible Markup Language (XML) 1.0
</locator>
```

سواء كان العميل قادراً على استرداد جزء واحد فقط خاص بالمستند وتكون بروتوكول معتمد. لا يمكن لأغلب مستعرضات الويب والخادم الحالية أن تكون قادرة على حمل المطالب المذكورة.والتي تتضمنها مؤشرات XML. ومع ذلك، يمكن أن يكون ذلك نافعاً للبروتوكولات العادية والتي تستخدم XML كآليات متنقلة تحتها خط.

مصطلحات الموقع المطلق

يتم بناء مؤشرات XML من مصطلحات الموقع، ويقوم كل مصطلح موقع بوصف نقطة في المستند المستهدف، وعامةً يكون هناك علاقة مع بعض النقط الأخرى المعروفة جيداً مثل البداية الخاصة بالمستند أو مصطلح موقع آخر، كما يتم إعطاء النوع الخاص بمصطلح الموقع بواسطة لوحة المفاتيح مثل id(), root() أو child().

تقوم بعض مصطلحات المواقع بأخذ المناقشات بين علامتي الجملة المعترضة، ومن أجل إظهار النقطة، يجب أن يكون لديك مثال واقعي في عقلك، وتمثل تعليمات البرمجة ١٧-١ مستند صالح بسيط والذي يجب أن يكون مفسراً لنفسه، وهو يحتوي على معلومات عن عائلتين مرتبطين وإعطاء هاتين العائلتين، ويكون مصدر العنصر هو FAMLY TREE ويمكن أن يحتوي FAMLY TREE على والعناصر Person و Famly، كما يحتوي كلاً من العناصر Famly و Person على السمة المطلوبة ID، وتحتوي الأشخاص على الاسم ويوم الميلاد ويوم الوفاه، بينما تحتوي العائلات على الزوج والزوجة الأطفال إن وجد، وتتم الإشارة إلى الأشخاص الفرديين في العائلة بواسطة الإشارة إلى الرقم المعرف الخاص بهم، وربما يتم حذف أي عنصر تابع من أي عنصر.

المرجع وسوف يتم إعادة التحدث عن تطبيق XML في الفصل ٢٣ في "تطبيـــق تصميـــم XML جديد".

تعليمات البرمجة ١٧-١: شجرة العائلة

```
<?xml version="1.0"?>
<!DOCTYPE FAMILYTREE [

 <!ELEMENT FAMILYTREE (PERSON | FAMILY)*>

 <!- PERSON elements ->
 <!ELEMENT PERSON (NAME*, BORN*, DIED*, SPOUSE*)>
 <!ATTLIST PERSON
  ID      ID     #REQUIRED
  FATHER  CDATA  #IMPLIED
  MOTHER  CDATA  #IMPLIED
 >
 <!ELEMENT NAME (#PCDATA)>
 <!ELEMENT BORN (#PCDATA)>
 <!ELEMENT DIED  (#PCDATA)>
 <!ELEMENT SPOUSE EMPTY>
 <!ATTLIST SPOUSE IDREF IDREF #REQUIRED>

 <!-FAMILY->
 <!ELEMENT FAMILY (HUSBAND?, WIFE?, CHILD*) >
 <!ATTLIST FAMILY ID ID #REQUIRED>

 <!ELEMENT HUSBAND EMPTY>
 <!ATTLIST HUSBAND IDREF IDREF #REQUIRED>
 <!ELEMENT WIFE EMPTY>
 <!ATTLIST WIFE IDREF IDREF #REQUIRED>
 <!ELEMENT CHILD EMPTY>
 <!ATTLIST CHILD IDREF IDREF #REQUIRED>

]>
```

```
<FAMILYTREE>

 <PERSON ID="p1">
  <NAME>Domeniquette Celeste Baudean</NAME>
  <BORN>11 Feb 1858</BORN>
  <DIED>12 Apr 1898</DIED>
  <SPOUSE IDREF="p2"/>
 </PERSON>

 <PERSON ID="p2">
  <NAME>Jean Francois Bellau</NAME>
  <SPOUSE IDREF="p1"/>
 </PERSON>

 <PERSON ID="p3" FATHER="p2" MOTHER="p1">
  <NAME>Elodie Bellau</NAME>
  <BORN>11 Feb 1858</BORN>
<DIED>12 Apr 1898</DIED>
  <SPOUSE IDREF="p4"/>
 </PERSON>

 <PERSON ID="p4" FATHER="p2" MOTHER="p1">
  <NAME>John P. Muller</NAME>
  <SPOUSE IDREF="p3"/>
 </PERSON>

 <PERSON ID="p7">
  <NAME>Adolf Eno</NAME>
  <SPOUSE IDREF="p6"/>
 </PERSON>

 <PERSON ID="p6" FATHER="p2" MOTHER="p1">
  <NAME>Maria Bellau</NAME>
  <SPOUSE IDREF="p7"/>
 </PERSON>
```

```
<PERSON ID="p5" FATHER="p2" MOTHER="p1">
  <NAME>Eugene Bellau</NAME>
</PERSON>

<PERSON ID="p8" FATHER="p2" MOTHER="p1">
  <NAME>Louise Pauline Bellau</NAME>
  <BORN>29 Oct 1868</BORN>
  <DIED>11 May 1879</DIED>
  <SPOUSE IDREF="p9"/>
</PERSON>

<PERSON ID="p9">
  <NAME>Charles Walter Harold</NAME>
  <BORN>about 1861</BORN>
  <DIED>about 1938</DIED>
  <SPOUSE IDREF="p8"/>
</PERSON>

<PERSON ID="p10" FATHER="p2" MOTHER="p1">
  <NAME>Victor Joseph Bellau</NAME>
  <SPOUSE IDREF="p11"/>
</PERSON>

<PERSON ID="p11">
  <NAME>Ellen Gilmore</NAME>
  <SPOUSE IDREF="p10"/>
</PERSON>

<PERSON ID="p12" FATHER="p2" MOTHER="p1">
  <NAME>Honore Bellau</NAME>
</PERSON>

<FAMILY ID="f1">
  <HUSBAND IDREF="p2"/>
  <WIFE IDREF="p1"/>
  <CHILD IDREF="p3"/>
```

```
    <CHILD IDREF="p5"/>
    <CHILD IDREF="p6"/>
    <CHILD IDREF="p8"/>
    <CHILD IDREF="p10"/>
    <CHILD IDREF="p12"/>
  </FAMILY>

  <FAMILY ID="f2">
    <HUSBAND IDREF="p7"/>
    <WIFE IDREF="p6"/>
  </FAMILY>

</FAMILYTREE>
```

من المفترض أن يتم تقديم هذا المستند عند

URL http://www.theharolds.com/genealogy.xml في القسم الأقدم وذلك لا يكـون URL حقيقي، ولكن الشيء الذي يجب التشديد عليه هو اختيار الأجزاء الفردية أكثر من المسـتند ككل.

id ()

إن مصطلح الموقع () id هو واحد من أبسط مصطلحات الموقع وأكثرها نفعاً، حيث يقوم باختيار العناصر الخاصة بالمستند والتي لديها قيمة النوع ID مع قيمة محددة. فعلى سبيل المثال، اعتبـار http://www.theharolds.com/genealogy. xml#id URL وإذا نظرت إلـى تعليمـات البرمجة ١-١٧، فسوف تجد هذا الفصل.

```
<PERSON ID="p12" FATHER="p2" MOTHER="p1">
  <NAME>Honore Bellau</NAME>
</PERSON>
```

بسبب أن سمة النوع ID استثنائية، فلا توجد عناصر أخرى تماثل مؤشرات X، ومع ذلك فإن http://www.theharolds.com/genealogy. xml#id(p12s). يجب أن تشير إلى العنصر PERSON الخاص بهونز بيلوز، لاحظ أن مؤشرات XML تختار العناصر الكاملة والتي تشـير إليها وتكون مشتملة على كل التوابع الخاصة بها وليس فقط رمز البداية.

إن النقص في مصطلح الموقع () ID يتمثل في أنه يتطلب مساعدات في المستند المسـتهدف، فإذا كان العنصر الذي تريد الإشارة إليها لا تحتوي على سمة النوع ID فيعتبر ذلك خط سـيئ،

وإذا كانت بعض العناصر في المستند تحتوي على سمة النوع ID، فيكون لديك القدرة علـــى أن تشير إلى واحد منهم وأن تستخدم مؤشرات XML القياسية "سوف تتم مناقشتها في الفصل القـــادم" للإشارة إلى الشيء الذي تريده فعلاً. وعلى الرغم من ذلك، تكون سمة النوع ID الأفضل عندمـــا يتم التحكم في كل من المستندات المستخدمة ومستندات الربط. لذلك يمكنك التأكيد على أن الرقـــم المعرف يماثل الارتباطات حتى إذا كان المستند يتطور أو يتغير عبر الوقت.

وفي بعض الحالات، مثل أن يكون المستند بدون DTD والمستند المستهدف ربما لا يحتـــوي على أي من سمات النوع ID، وعلى الرغم من ذلك قد يكون لديها السمات المسماة ID، وفي هذه الحالة، ربما يقوم التطبيق بمحاولة تضمين أي العناصر الذي تمت الإشارة إليه، وعامـــةً، تقـــوم باختيار العنصر الأول في المستند على السمة الخاصة بأي نوع واسم تماثل قيمة ID المطلـــوب ومن ناحية أخرى، يكون التطبيق حراً في عدم اختيار أي عنصر.

Root()

يشير مصطلح الموقع () Root إلى عنصر الجذر الخاص بالمستند، وهي لا تأخذ أي مناقشـــة، فعلـــى ســـبيل المثـــال، يتـــم وصـــف عنصـــر الجـــذر الخـــاص بلغـــة XML 1.0 فـــي http://www.w3.org/TR/REC-xml ويكون هو Spec، وهكذا لاختيارها يمكنـــك اســـتخدام URL:

http://www.w3.org/TR/REC-xml#root()

يكون مصطلح الموقع () Root مفيد مبدئياً في مؤشرات XML المركبة كأساس للبداية. فـــي المواقع، إذا كان لا يتم اشتمال مصطلح الموقع المطلق في مصطلح الموقع المركب، بحيث يكون () Root فرضياً، ومع ذلك، يمكن استخدام () Root أيضاً في اختيار المستند الكامل بلغـــة URL والتي تستخدم للدلالة على أن جزء واحد فقط قد تم تحميله بشكل طبيعي، فعلى سبيل المثال:

http://www.w3.org/TR/1999/REC-xml-names-19990114/xml-names.xml|root()

html()

يقوم مصطلح الموقع ()html باختيار الاسم المعتمد الخاص بمستندات HTML، ويكـــون لديـــها وسائط مفردة، وهي الاسم الخاص بالمعتمد الذي تشير إليه، فعلى سبيل المثـــال، يوجـــد الاســـم المعتمد التالي في الملف: http://metalab.unc.edu/xml/:

```
<a name="quote"><font color="#AA0000">Quote of the
Day</font></a>
```

ويكون مؤشر XML الذي يشير إلى هذا العنصر:

http://metalab.unc.edu/xml#html(quote)

يوجد مصطلح الموقع () html الأولى من أجل التوافق الخلفي والذي يسمح لمؤشـوات XML بالإشارة إلى مستندات HTML، وقد يستخدم الاسم المعتمد في مستندات XML في إمداد كل قيـم السمات المقتبسة وإعلان العنصر A والسمات الخاصة به في DTD ومقابلة كل المعـايير جيـدة التكوين الأخرى، وعامةً، على أي حال، يكون لدى XML وسائل أفضل في التـي لـدى الاسـم المعتمد لتحديد الموقع.

مصطلحات الموقع النسبي

تمثل html, id, root مصطلحات الموقع المطلق، ويمكن لمصطلحات الموقع المطلـق إيجـاد عناصر معينة في المستند، بغض النظر عن ماذا يوجد أيضاً داخل المستند، ومع ذلك، فإنه فـي الشائع أن تريد إيجاد العنصر الأول الخاص بالنوع المعطى، والعنصر الأخير الخاص بالمسـتند المعطى والتابع الأول لنوع معين والعنصر الخاص بالنوع المعطى وكل العناصر الخاصة بالنوع المعطى أو ما شابه ذلك، وتتم هذه العملية بواسطة تلعق مصطلح الموقع النسبي المطلق لتكويـن المركب. ويكون مؤشر XML أكثر عموماً عندما يتم إتباع مصطلح الموقـع المطلـق Locator الفردي بواسطة أي عدد في مصطلحات الموقع النسبي.

ويكون كل مصطلح في تعليمات البرمجة نسبياً بالنسبة للمصطلح الذي يسبقه فيما عدا بالنسبة لمصطلح الموقع المطلق الأول، ويتم فصل المصطلحات الموجودة في تعليمات البرمجة بواسـطة المدة.

فعلى سبيل المثال، انظر إلى مستند شجرة العائلة في تعليمات البرمجـة ١٧-١ يقـوم هـذا التجزيء باختيار العنصر الأول NAME الخاص بالعنصر السادس Person في عنصر الجذر.

http://www.theharolds.com/genealogy.xml#root().child(6,PERSON).child (1,NAME)

In this example, that's <NAME>Maria Bellau</NAME>.

في هذا المثال، <NAME>Maria Bellau</NAME>

فعلى سبيل مثال آخر، افترض أنك تريد الارتباط بـالعنصر Name الخـاص بـالمصطلح Domeniquette Celeste Baudean، ويكون أسهل طريق لفعل ذلك هـو تحديـد العنصـر Person بواسطة ID, p1، وبعد ذلك قم باستخدام المصطلح الموقع النسبي ()child للإشارة إلى العنصر التابع الأول "والوحيد" Name، مثل ذلك:

http://www.theharolds.com/genealogy.xml#id(p1)

وقد تم الإخبار عن URL لتصل إلى المستند

http://www.theharolds.com/genealogy لإيجاد عنصر الجذر الخاص بها وبعد ذلك إيجاد العنصر مع ID pl ثم اختار التابع الأول Name الخاص بها.

وعلى الرغم من اشتمال geneaology.xml للسمة ID الخاصة بأغلب العناصر وعلى الرغم من ملاءمتهم فإنه لا يتطلب منهم الارتباط داخل المستند. كما يمكنك اختيار أي عنصر من المستند يبسط No بواسطة العد التنازلي من عناصر الجذر. ويمكن عد Person واحد تنازلياً في الجذر، وبعد ذلك عد Name واحد تنازلياً في ذلك الآن ماري بلوز هي أول شخص في المستند. وبذلك يتم URL الآتي:

```
http://www.theharolds.com/genealogy.xml#root().child(1,
PERSON).child(1,NAME)
```

وقد تم إخبار URL أيضاً بالتالي إلى المستند لإيجاد عناصر الجذر الخاصة بها، وبعد ذلك إيجاد العنصر الأول Person والذي يمثل التابع الحالي الخاص بعنصر الجذر، وبعد ذلك إيجاد العنصر الأول Name الخاص بها.

فإذا لم يتم تضمين المصطلح الموقع المطلق في Xpointer، فسوف يعد ذلك افتراض root ()، فعلى سبيل المثال، كتابة المثال السابق على نحو أكثر إدماجاً، مثل ذلك:

```
http://www.theharolds.com/genealogy.xml#child(1,PERSON).child
(1,NAME)
```

•ويمكنك ضغط ذلك أكثر لواسطة حذف مصطلح الموقع Child الثاني "على الرغم من انه ليس الوسيط الخاص بها". فعلى سبيل المثال:

```
http://www.theharolds.com/genealogy.xml#child(1,PERSON).(1,NAME)
```

عندما يتم حذف المصطلح بهذه الطريقة، فإنه من المفترض أن يكون مشابهاً للمصطلح السابق، لأنه لا يوجد مصطلح أمام (1, Name) فإنه من المفترض أن تصبح مشابهة للمصطلح السابق، إلا وهو Child.

كما يوجد اختيارات فعالة في التقنيات الأخرى، والتي سوف تناقش أدناه وفي الواقع، يوجد سبعة مصطلحات الموقع النسبي بالإضافة إلى () Child، والتي سوف يتم وضعهم في تعليمات البرمجة ١٧-١، ويقوم كل خادم باختيار مجموعة أولية معينة الخاصة بالعناصر في المستندات. فعلى سبيل المثال، فقد تم اختيار مصطلح الموقع النسبي Following من العناصر التي جاءت بعد عنصر المصدر، كما تم اختيار مصطلح الموقع النسبي Preceding من العناصر التي جاءت بعد عناصر المصدر.

الجدول ١٧-١

مصطلحات الموقع النسبي

المصطلح	المعنى
Child	يتم اختياره من التوابع الحالية الخاصة بعناصر المصدر
Descendant	يتم اختياره من أي من المحتويات أو عناصر التوابع الخاصة بعنصر المصدر.
Ancestor	يتم اختياره من العناصر التي تحتوي على عنصر المصدر
Preceding	يتم اختياره من العناصر التي تسبق عنصر المصدر
Following	يتم اختياره من العناصر التي تلي عنصر المصدر
Psibling	يتم اختياره من العناصر المتقاربة التي تسبق عنصر المصدر
Fsibling	يتم اختياره من العناصر المتقاربة التي تلي عنصر المصدر.

بما إن مصطلح الموقع النسبي وحيداً يكون عامةً غير كافياً لتحديد أي العناصر يجب أن يشار إليها، وتقوم الوسطاء الإضافيين بتمرير هذا التعريف الإضافي الخاص بالعنصر المستهدف بواسطة أرقام الأمثلة ونوع العقد والسمات، ويكون الوسيط الممكن مماثلاً لكل مفاتيح الموقع السبعة، حيث يقومون باستكشاف أكثر تفصيلاً في وسيط الموقع النسبي الخاص بالقسم أدناه.

التابع

يتم اختيار مصطلح الموقع النسبي Child من التوابع المباشرة الخاصة بعنصر المصدر، فعلى سبيل المثال، باعتبار URL

http://www.theharolds.com/genealogy.xml#root().child(6,NAME)

توجد هذه النقط ولأن لا يوجد العنصر Name في المستند المباشر آلا وهو التوابع المباشرة الخاصة بالجذر.ويوجد رزمة من عناصر Name والتي تمثل توابع غير مباشرة، فإذا كنت تريد الإشارة إلى ذلك، فإنه يجب استخدام عنصر الموقع النسبي descendant بدلاً من Child.

الانحدار

يقوم مصطلح الموقع النسبي descendant بالبحث خلال كل الانحدارات الخاصة بالمصدر، وليس فقط التوابع المباشرة، فعلى سبيل المثال، تقوم root().descendant(3,BORN) باختيار

العنصر الثالث Born مواجهاً للبحث الأول العميق الخاص بمستند الشجرة. "يتم تعريف العمق الأول على أنه الآخر الذي تحصل عليه إذا كنت تقرأ ببساطة خلال المستند الخاص بلغة XML من الأعلى إلى الأسفل" وفي تعليمات البرمجة ١٧-١ والتي تختار الميلاد بلويس بولن بيلوز.

<BORN>29 Oct 1868</BORN>

الأصل

يقوم مصطلح الموقع النسبي ancestors بالبحث خلال كل الأصول الخاصة بالمصدر، مبتدأ بالأقرب، حتى يجد العنصر المطلوب فعلى سبيل المثال، يقوم root().descendant(2,BORN).ancestor(1) باختيار العنصر Person، والذي يحتوي العنصر الثاني Born ففي هذا المثال، نقوم باختيار العنصر Person الخاص بأليودي بيلوز.

السابق

يقوم مصطلح الموقع النسبي Preceding بالبحث خلال كل العناصر التي توجد بعد عنصر المصدر، ولا يكون لدى عنصر محدد الموقع Preceding أي تقرير بالنسبة للتسلسل الهرمي، ففي أول مرة يتم مواجهة العنصر الخاص برمز البداية ورمز النهاية أو رمز الفراغ والتي تقوم بعد هذا الفصل، فعلى سبيل المثال، يجب اعتبار هذا القانون:

```
root().descendant(3,BORN).preceding(5)
```

ويقوم ذلك بالأخبار بالذهاب إلى عيد ميلاد بولين بوليز <BORN>29 Oct 1868</BORN> وبعد ذلك يتم تحريك خمسة عناصر إلى الخلف، ويستقر ذلك في الغنصر Person الخاص بماريا بوليز.

اللاحق

يقوم مصطلح الموقع النسبي Following بالبحث خلال كل العناصر التي توجد بعد عنصر المصدر في المستند، مثل preceding, following ليس لديها أي اعتبارات بالنسبة للتسلسل الهرمي، ففي أول مرة تواجب العناصر الخاصة برمز البداية ورمز النهاية أو رمز الفراغ، فأنها تقوم بعد هذه العناصر، فعلى سبيل المثال، فباعتبارها هذا القانون:

```
root().descendant(2,BORN).following(5)
```

ويقوم ذلك بالأخبار بالتوجه إلى عيد ميلاد إيلودي بيلوزر <BORN>11 Feb 1858</BORN> وبعد ذلك تحريك خمس عناصر إلى الأمام، حيث يقوم ذلك بالاستقرار في

العنصر Name الخاص بمولر إلا هو <NAME>John P. Muller</NAME> بعد المرور خلال العنصر DIED الخاص بالعنصر إيلوت بيلوز والعنصر Spouse الخاص ببيلوز إيلوت والعنصر Person الخاص ببيلوز إيلوت والعنصر Person الخاص بجون بر مولر، في هذا الأمر.

Psibling

يقوم مصطلح الموقع النسبي Psibling باختيار العنصر الذي يسبق عنصر المصدر في نفس العنصر الأصلي، فعلى سبيل المثال يقوم root().descendant(2, BORN).psibling(1) باختيار العنصر Name الخاص بأليود بيلود وهو <NAME>Elodie Bellau</NAME>. root().descendant(2, BORN) لا تشير إلى شيء لأنه يوجد قريب واحد فقط خاص بالعنصر Name الخاص باليودي بيلور قبلها.

Fsibling

يقوم مصطلح الموقع النسبي Fsibling باختيار العنصر الذي يلي المصدر في نفس العنصر الأصلي، فعلى سبيل المثال، يقوم root().descendant(2,born). fsibling(1) باختيار العنصر DIED الخاص بأليودي إيلوت. إلا وهي <DIED>12 Apr 1898</DIED>. root().descendant(2,born).fsibling(3) ولا يشير إلى شيء يوجد عنصرين قريبين فقط يلون العنصر Name الخاص بأليودي بيولز.

وسيط مصطلح الموقع النسبي

يبدأ كل مصطلح الموقع النسبي عند مكان معين في المستند يسمى مصدر الموقع، وعامةً، يتم تضمين مصدر الموقع بواسطة مصطلح الموقع المطلق "أو الجذر إذا لم يتم تحديد مصطلح مطلق"، يجب بعد ذلك البحث للأمام أو للخلف في المستند للبحث عن أول مطابق يقابل المعايير المحددة.

يتم إعطاء المعايير كقائمة في الوسطاء إلى مصطلح الموقع النسبي، وقد يشتمل ذلك على البحث عن الأرقام الخاصة بالعناصر الأمام أو الخلف والبحث عن أنواع الأشياء "عناصر، تعليق تعليمات المعالجة وغير ذلك" و/أو والبحث عن القيمة الخاصة بالسمات يتم إعطاء ذلك في هذا الترتيب:

١- الأعداد

٢- النوع

٣- السمة

ويكون العدد الصحيح إيجابياً أو سلبياً عندما يعد للأمام أو الخلف من مصدر الموقع، ويعرف النوع على أنه نوع خاص بالأشياء التي تعد. بينما تعرف السمة على أنها قائمة خاصة بأسماء السمات والقيم التي يجب مماثلتها، ويمكن لمصطلح الموقع النسبي أن يحتوي على عدد وقد يكون رقم ونوع أو نوع فقط أو رقم فقط وقائمة من السمات.

ويتم تباعد الوسطاء الذين تم تقديرهم بواسطة فصله وبدون مسافة فارغة فعلى سبيل المثال:

child(1,PERSON,FATHER,p2)

وتكون متطلبات المسافة الفارغة غير عادية، فبوجودها يمكن لمؤشرات X بسهولة التعلق بالنهاية بالمصطلح URL. فعلى سبيل المثال:

http://www.theharolds.com/genealogy.xml#child(1,PERSON,FATHER,p2
)

فإذا كان مسموحاً للمسافة الفارغة، فسوف يصبح URL مثل ذلك:

http://www.theharolds.com/genealogy.xml#child(1,%20PERSON,
%20FATHER,%20p2)

ولمعظم الأجزاء، سوف يتم تطبيق نفس التركيب في كل مصطلحات الموقع النسبي السبعة.

الاختيار بواسطة العدد

إن أسهل تركيب بالاختيار يكون بواسطة الإعداد، ويكون الوسيط الأول بالنسبة لمصطلح الموقع النسبي هو الفهرس الخاص بالقاعدة التي تشير إليها، وتقوم الأعداد الإيجابية بالعد للأمام في المستند، بينما تعد الأعداد السلبية إلى الغلق، كما يمكن أيضاً استخدام الكلمة الأساسية All للإشارة إلى كل العقد التي تماثل الحال.

العدد الأمامي

فعلى سبيل المثال، في تعليمات البرمجة١٧-١ يمثل Family Tree الجذر، وهو يحتوي على ١٤ تابع مباشر و١٢ عنصر Person وعنصرين في العنصر Famly في هذا الترتيب:

http://www.theharolds.com/genealogy.xml#root().child(1)
http://www.theharolds.com/genealogy.xml#root().child(2)
http://www.theharolds.com/genealogy.xml#root().child(3)
http://www.theharolds.com/genealogy.xml#root().child(4)
http://www.theharolds.com/genealogy.xml#root().child(5)

http://www.theharolds.com/genealogy.xml#root().child(6)
http://www.theharolds.com/genealogy.xml#root().child(7)
http://www.theharolds.com/genealogy.xml#root().child(8)
http://www.theharolds.com/genealogy.xml#root().child(9)
http://www.theharolds.com/genealogy.xml#root().child(10)
http://www.theharolds.com/genealogy.xml#root().child(11)
http://www.theharolds.com/genealogy.xml#root().child(12)
http://www.theharolds.com/genealogy.xml#root().child(13)
http://www.theharolds.com/genealogy.xml#root().child(14)

لا تشير الأعداد العظيمة مثل http://www.theharolds.com/genealogy.xml#root().child(15) إلى أي مكان، ولكنهم يتبعون فقط URLs.

يمكن استخدام Descendant بدلاً من Child، لعد كل العناصر في المستند، وليس فقط التوابع المباشرة الخاصة بالجذر، كما يظهر الجدول ١٧-٢ الأربعة الأولى من Descendant الخاصة بمؤشرات X لتعليمات البرمجة ١٧-١ والذي يشيرون إليه، ولاحظ على وجه الخصوص أن root ().descendant (1) تشير إلى العنصر الكامل Person بالإضافة إلى التوابع الخاصة بها، وليس فقط رمز البداية

الجدول ١٧-٢

الأربعة الأولى من الانحدار الخاصة بالجذر

مؤشرات X	تشير إلى
<PERSON ID="p1">	<NAME>Domeniquette Celeste Baudean</NAME> <BORN>11 Feb 1858</BORN> <DIED>12 Apr 1898</DIED> <SPOUSE IDREF="p2"/> </PERSON>
root().descendant(2)	<NAME>Domeniquette Celeste Baudean</NAME>
root().descendant(3)	<BORN>11 Feb 1858</BORN>
root().descendant(4)	<DIED>12 Apr 1898</DIED>

الأرقام الخلفية

تقوم الأرقام السلبية بتمكينك من التحرك للخلف من العنصر الحالي إلى البند الذي تشير إليه، ففي حالة Descendant, Child يقومون بالعد من الخلف من رمز النهاية بينما في حالـــة التحــرك للأمام يبدأ من رمز البداية، فعلى سبيل المثال، تقوم مؤشرات X باختيار العناصر التـــي تسـبق مباشرة العناصر الخاصة: IDF1

http://www.theharolds.com/genealogy.xml#id(f1).following(-1)

ففي هذا المثال،يكون العنصر Person خاصاً بهونز بيلو، ومع ذلك سوف تكون ارتباطـــاتك صحيحة إذا أمكنك تجنب الأرقام السلبية عندما يمكنك ذلك وتقوم باستخدام محدد بديل.فعلى سبيل المثال، يقوم ذلك باختيار نفس العناصر:

http://www.theharolds.com/genealogy.xml#id(f1).preceding(1)

تشير الأرقام السلبية الموجود في الشجرة المبينة على المحددين مثـل Descendant, Child على انه يجب العدد من النهاية الخاصة بالأصل بدلاً من البداية، فعلى سبيل المثال، يقـــوم ذلـك بالإشارة إلى العنصر Person في المستند:

وذلك يشير إلى العنصر قبل الأخير Person في المستند.

http://www.theharolds.com/genealogy.xml#root().child(-1,person)

يعرض الجدول ١٧-٣ الأربعة الأخيرة من مؤشرات Descendant X الخاصة بتعليمـــات البرمجة ١-١٧ وما الذي يشيرون إليه، لاحظ الترتيب الذي تم به إدخال العناصر يكون موجـوداً الآن بواسطة رمز النهاية أكثر من رمز البداية.

الجدول ١٧-٣
الأربعة عناصر المنحدرة الخاصة بالجذر

مؤشرات X	تشير إلي
root().descendant(1)	<FAMILY ID="f2"> <HUSBAND IDREF="p7"/> <WIFE IDREF="p6"/> </FAMILY>
<WIFE IDREF="p6"/>	root().descendant(2)
<HUSBAND IDREF="p7"/>	root().descendant(3)

الجدول ١٧-٣

الأربعة عناصر المنحدرة الخاصة بالجذر

مؤشرات X	تشير إلي
Root().descendant(4) <HUSBAND IDREF="p2"/>	<FAMILY ID="f1">

```
<WIFE IDREF="p1"/>
<CHILD IDREF="p3"/>
<CHILD IDREF="p5"/>
<CHILD IDREF="p6"/>
<CHILD IDREF="p8"/>
<CHILD IDREF="p10"/>
<CHILD IDREF="p12"/>
</FAMILY>
```

الكل

بالإضافة إلى تحديد العدد للاختيار، يمكن استخدام الكلمة الأساسية all، ويقوم ذلك بالإشارة إلى كل العقد التي تماثل الحالة. فعلى سبيل المثال، يشير هذا القانون إلى كل التوابع الخاصة بالمستند المحتوي على: IDF1

```
http://www.theharolds.com/genealogy.xml#id(f1).child(all)
```

وبعبارة أخرى، فإنه يشير إلى:

```
<HUSBAND IDREF="p2"/>
<WIFE IDREF="p1"/>
<CHILD IDREF="p3"/>
<CHILD IDREF="p5"/>
<CHILD IDREF="p6"/>
<CHILD IDREF="p8"/>
<CHILD IDREF="p10"/>
<CHILD IDREF="p12"/>
```

الاختيار بواسطة نوع العقدة

يقوم القانون أعلاه باختيار عنصر معين من المستند، ومع ذلك، ففي بعض الأحيان تحتاج إلى اختيار WIFE الخامس أو Person الثالث مع تجاهل العناصر الخاصة بالأنواع الأخرى، اختيار

ذلك بواسطة مثال عدد وحيد فأنه ذلك يميل إلى الخطأ إذا تغير المستند أن إضافة أو حذف عنصر مفرد في مكان غير مناسب ممكن أن يؤدي إلى فقدان محاذاة كل الارتباطات التي تعتمد فقط على مثال العدد.

ومن حين لآخر فقد تحتاج إلى اختيار تعليمات المعالجة وقسم Cdata أو نص معين حالته الأولى في المستند، ويمكن إكمال ذلك بواسطة إضافة وسيط ثاني إلى مصطلح الموقع النسبي بعد العدد والذي يحدد أي العقد تقوم بعدها "وضمنياً" أي العقد تقوم بتجاهلها. ويمكن أن يكون ذلك الاسم الخاص بالعنصر والذي تريد الإشارة إليه أو واحد من الست مفاتيح الموضوعة في القائمة في الجدول ١٧-٤.

الجدول ١٧-٤

الوسيط الثاني الممكن الخاص بمصطلحات الموقع النسبي

النوع	التماثل
#element	أي عنصر
#pi	أي تعليمات معالجة
#comment	أي تعليق
#text	أي صفة بيانات غير ترميزية
#cdata	أقسام CDATA
#all	كل ما جاء أعلاه
Name	العناصر الخاصة بالأسماء المحددة

وتشمل أغلب قوانين الاختيار نوع العنصر الذي تم البحث عنه، ويمكنك الآن مشاهدة الأمثلة عندما تقوم ROOT () Child (6, Person) باختيار التابع السادس person الخاص بالجذر وقد يشير ذلك إلى شخص غير صحيح، إن أتم إضافة أو حذف العنصر Person ولكن في النهاية يكون العنصر Person بدلاً من أي شيء أخر مثل Family.

كما يمكنك أيضا أن تضف النوع فقط وتحذف مثال الرقم "وليس الفاصلة". فعلى سبيل المثال، تقوم URL باختيار كل العناصر Person في المستند بغض النظر عن الموقع.

http://www.theharolds.com/genealogy.xml#root().child(,PERSON)

يجب أن توجه انتباهك على وجه الخصوص إلى الفاصلة الوحيدة أمام Person. ويتم تطلب ذلك بواسطة قواعدBNF في النسخة الحالية الخاصة بمواصفات مؤشرات X، فوجودها يجعل منها

أكثر سهولة للبرامج لتعريب مؤشرات X، حتى إذا جعلت ذلك صعباً علـــى الأشـــخاص الذيـــن يريدون قراءة مؤشرات X.

ماذا تفعل التطبيقات عندما تكون كل عناصر Person مستهدفة يعود ذلك إلـــى التطبيقـــات، فعامةً، يوجد شيء أكثر تعقيداً في التحميل في ذاته ووصفه في العنصر المستهدف يتم اقتراحــــه عندما يوجد أكثر من عنصر واحد مستهدف، فإذا كانت التطبيقات تستخدم هذا الجزء لتحديـــد أي أجزاء المستند يجب تحميلها وبعد ذلك يتم تحميل كل العناصر الخاصة بالعنصر المحدد.

ومع ذلك، فإن ذلك غير عادي، ففي أغلب الأوقات يتم الاختيار بواسطة نوع يستخدم فقط مع التحديد أكثر بالنسبة للعناصر المختارة حتى يبقى واحد منها فقط مستهدف.

الاسم

إن الاستخدام الأكثر شيوعاً للوسيط الثاني الخاص بمصطلح الموقع النسبي هو إعطاء اسم خاص بنوع العنصر، فعلى سبيل المثال، أفترض إنك تريد الإشارة إلى العنصر الأول Famly إلا وهـــو التابع الخاص بعنصر الجذر، ولكنك لا تعرف كيف تم اختلاطها مع العنصـــــر Person، وهـــذا القانون يتم ذلك:

http://www.theharolds.com/genealogy.xml#root().child(1,FAMILY)

ويكون ذلك فعال عملياً، عندما تسلسل قوانين التطبيق، فعلى سبيل المثال، فإن ذلك يشير إلـــى العنصر الثاني Child الخاص بالعنصر الأول Famly:

http://www.theharolds.com/genealogy.xml#root().child(1,FAMILY).child (2,CHILD)

وفي الواقع، يكون تحديد النوع الخاص بالعنصر الذي قمت باختياره أكثر شيوعاً، أكثر مـــن عدم تحديده. ويكون ذلك على وجه الخصوص صحيحاً بالنسبة لمصطلح الموقع النسبي والذي لا يتبع التسلسل الهرمي مثل Following وPrecediny.

العنصر

إذا كان الوسيط الثاني غير محدد، تكون العناصر بذلك متماثلة، ولكن تكون تعليمـــات المعالجـــة والتعليقات والقسم CDATA صفات البيانات وغير ذلك غير متماثلة. ويمكنك تكرار هذا الســـلوك مع الكلمة الإرشادية #element مثل الوسيط الثاني، فعلى سبيل المثال، حيـــث يكـــون هذيـــن URLs متماثلان:

http://www.theharolds.com/genealogy.xml#id(f2).preceding(1)
http://www.theharolds.com/genealogy.xml#id(f2).preceding (1,#element)

ويكون السبب الرئيسي لاستخدام element# هو أنه يمكنك من استخدام الوسيط حتى يكــون متماثلاً ضد السمات.

#text

يقوم الوسيط text# باختيار النص الخام من داخل العنصر، ويكــون اســتخدامها الشــائع مــع المحتويات المختلطة، فعلى سبيل المثال، باعتبار هــذا العنصــر Citation الخــاص بتعليمــات البرمجة ٣-١٢ في الفصل ١٢.

```
<CITATION CLASS="TURING" ID="C2">
  <AUTHOR>Turing, Alan M.</AUTHOR>
  "<TITLE>On Computable Numbers,
    With an Application to the Entscheidungs-problem</TITLE>"
  <JOURNAL>
    Proceedings of the London Mathematical Society</JOURNAL>,
  <SERIES>Series 2</SERIES>,
  <VOLUME>42</VOLUME>
  (<YEAR>1936</YEAR>):
  <PAGES>230-65</PAGES>.
</CITATION>
```

وتشير مؤشرات X التالية إلى علامتا الاقتباس قبل العنصر Title.

```
id(C2).child(2,#text)
```

وتكون عقدة الفصل الأول في هــذا الجــزء هــي مســافة الفــراغ بيــن <CITATION CLASS="TURING" ID=""> و<AUTHOR>، وفنياً، تقوم مؤشرات X بالإشارة إلى كــل النصوص بين <TITLE> و<AUTHOR>، والتي تشتمل على مسافة الفراغ وليس فقط علامتــا الاقتباس.



وتقوم مؤشرات X بالإشارة إلى عقد النص المخادعة، وأنصح بالابتعاد عنهم بقدر الإمكان، تماماً مثل تجنبك المحتويات المختلطة ومن الطبيعي، قد يكــون لديــك القدرة على فعل ذلك دائماً، وعلى وجه الخصوص إذا كنت تريد الإشــارة إلــى الأجزاء الخاصة بالمستند الذي تمت كتابته بواسطة مؤلف أخــر لا يتبــع هــذا التدريب الجيد.

لأن مؤشرات الأحرف لا تحتوي على العناصر التابع C فقد لا تكون مصطلحــات الموقــع النسبي الإضافي متعلقة بمؤشرات X والتي تتبع أحدهم وهو الذي يقوم باختيار عقد النص، ومنــذ عدم احتواء بيان الأحرف على السمات، فقد لا يتم استخدام وسطاء السمة بعد Text.

#cdata

يقوم الوسيط CDATA بتحديد أن القسم CDATA ويكون أكثر احتمالاً، النص الخاص بالقسم CDATA، يجب أن يتم اختياره، فعلى سبيل المثال، ويقوم مؤشر X بالإشارة إلى القسم الثاني CDATA في المستند:

```
root().following(2,#cdata)
```

ولعدم قدرة القسم CDATA على الحصول على توابع، فقد لا يمكن لمصطلح الموقع النسبي الإضافي القدرة على التعلق بمؤشرات X والتي تتبع أحدهم والتي تقوم باختيار القسم CDATA ومنذ عدم قدرة القسم CDATA على الحصول على سمات، فقد لا يتم استخدام وسيط السمات بعد #CDATA..

#pi

وفي مواقف نادرة قد تحتاج إلى اختيار تعليمات المعالجة أكثر من العنصر، وفي هذه الحالة، يمكنك استخدام PI# كوسيط ثاني بالنسبة لمصطلح الموقع، فعلى سبيل المثال، يقوم مؤشر X باختيار تعليمات المعالجة الثانية في المستند الخاص بالعنصر الثالث BEAN.

```
root().descendant(3,BEAN).child(2,#pi)
```

ولأن تعليمات المعالجة لا تحتوي على سمات أو عناصر، فلذلك لا يمكنك إضافة مصطلحات الموقع النسبي الإضافي بعد المصطلح الأول الذي يقوم باختيار تعليمات المعالجة، ومع ذلك، فإنه يمكنك أن تستخدم مصطلح الموقع String لاختيار الجزء الخاص بالنص الخاص بتعليمات المعالجة.

#comment

تقوم مؤشرات X بالإشارة إلي التعليق بطريقة مماثلة كثيراً للطريقة التي تتم بها الإشارة إلى تعليمات المعالجة، ويتم استخدام Comment# الواقعي كوسيط ثاني بالنسبة لمصطلح الموقع، فعلى سبيل المثال، يقوم مؤشر X بالإشارة إلى التعليق الثالث في تعليمات البرمجة ١-١٧.

```
http://www.theharolds.com/genealogy.xml#descendant(3,#comment)
```

ولأن التعليقات لا تحتوي على سمات أو عناصر، فلا يمكنك إضافة مصطلح الموقع النسبي الإضافي بعد المصطلح الأول الذي يقوم تعليمات المعالجة، كما يمكنك استخدام مصطلح الموقع String () لاختيار الجزء الخاص بتعليمات المعالجة.

#all

وفي الحالات النادرة جداً، فقد تحتاج إلى اختيار عقد معينة من المستند بغض النظر إذا كانت عناصر أو بيانات حروف خام أو تعليمات معالجة أو قسم CDATA أو تعليق، ولعمل ذلك أعتقد أن الشيء الوحيد الذي يجعلني أفكر في ذلك إذا كان يتم تكرار خلال كل العقد في المستند أو

العنصر، بواسطة استخدام all # توسيط ثاني لمصطلح الموقع النسبي، يمكنك تجاهل نوع الشيء الذي تقوم بمماثلته، فعلى سبيل المثال، انظر إلى هذا الجزء الخاص بتعليمات البرمجة ٣-١٢ في فصل ١٢:

```
<CITATION CLASS="TURING" ID="C3">
  <AUTHOR>Turing, Alan M.</AUTHOR>
  "<TITLE>Computing Machinery &amp; Intelligence</TITLE>"
  <JOURNAL>Mind</JOURNAL>
  <VOLUME>59</VOLUME>
  (<MONTH>October</MONTH>
  <YEAR>1950</YEAR>):
  <PAGES>433-60</PAGES>
</CITATION>
```

يضع الجدول ٥-١٧ مؤشرات X الأربعة والتي تبسط عد العقد التنازلي في العنصر Citation في القائمة X، كما أنها تقوم بوضع ما يشار إليه في القائمة بواسطة مؤشرات X.

الجدول ١٧-٥
عقد مؤشرات إكس الأربعة الأولى الخاصة بالعنصر Citation

مؤشرات X	تشير إلى
The whitespace between <CITATION CLASS="TURING" ID="C3"> and <AUTHOR>	id(C3).following(1,#all)
<AUTHOR>Turing, Alan M.</AUTHOR>	id(C3).following(2,#all)
Turing, Alan M.	id(C3).following(3,#all)
"	id(C3).following(4,#all)

الاختيار بواسطة السمات

يتم إضافة الوسيط الثالث والرابع إلى مصطلح الموقع النسبي للإشارة إلى العناصر بواسطة السمات، ويمثلان الوسيط الثالث اسم السمة، ويمكنك الوسيط الرابع قيمة السمة، فعلى سبيل المثال، لإيجاد العنصر الأول Person في المستند. http://www.theharolds.com/genealogy.xml والتي هي السمة Father يكون جاين فرانكوس بيلو (ID p2) ويمكن كتابة:

```
root().child(1,PERSON,FATHER,p2)
```

ويقوم هذا القانون باختيار العنصر الأول Person في المستند الذي يحتوي على قيمة العنصر الخاص P2 بغض النظر عن ما إذا كانت السمة تظهر على هيئة Father أو Mother أو ID أو شيء آخر.

```
root().child(all,#element,*,p2)
```

في تعليمات البرمجة ١-١٧ يمثل ذلك العنصر Person الخاص بجان فرانكوس بيلو.

بوضع الوسيط الرابع على هيئة علامة نجمية تدل على أن أي قيمة يسمح لها بالإضافة إلى القيم الافتراضية القراءة في الإعلان ATTLIST الخاص بالمصطلح DTD فعلى سبيل المثال، يقوم هذا القانون باختيار العنصر الأول في المستند والذي يحتوي على سمة Father.

```
root().child(1,#element,FATHER,*)
```

في تعليمات البرمجة ١٧-١، يمثل العنصر Person العنصر الخاص بأيلودي بيلوز.

كما يمكنك استخدام Implied# كوسيط رابع ليماثل ضد السمات التي ليس لديها قيم، سواء إذا كانت محددة أو افتراضية بصفة مباشرة. فعلى سبيل المثال، يجد هذا القانون السطر الأول Person لا يحتوي على السمة Father.

```
root().child(1,PERSON,FATHER,#IMPLIED)
```

في تعليمات البرمجة ١٧-١، ويمثل ذلك العنصر Person الخاص بالمصطلح Dom Etiquette.

يقوم وسيط السمة بالعمل فقط في مصطلحات الموقع النسبي التي تقوم باختيار العنصر، ويمكنك استخدامهم عندما يكون الوسيط الثاني ,#text,#cdata,#pi أو #comment لأن هذه العناصر لا تحتوي على سمات.

سلسلة مصطلحات الموقع

يكون اختيار العناصر المعينة في بعض الأحيان يكون جيداً للإشارة داخل مستندات XML جيدة التكوين، ومع ذلك، في حالة الاحتياج إلى الإشارة إلى بيانات غير خاصة بلغة XML وبيانات خاصة بلغة XML والتي بها مقدار كافي من النصوص غير الخاصة بلغة XML يتم تضمينها بواسطة القسم CDATA والتعليقات وتعليمات المعالجة أو بعض العناصر الأخرى، وفي هذه الحالة، قد تحتاج إلى الإشارة إلى نطاق معين خاص بالنص داخل المستند الذي لا يتم تفصيله في ترميز أي عنصر معين، ويمكنك استخدام سلسلة مصطلحات الموقع لعمل ذلك.

وتقوم سلسلة مصطلح الموقع بالإشارة إلى وجود سلسلة معينة، وما غيراً لأغلب مصطلحات الموقع، يمكن تسلسله مصطلح الموقع الإشارة إلى الموقع داخل التغليف و CDATA وما شبه

ذلك. فعلى سبيل المثال يقوم هذا الجزء بإيجاد أول وجود لسلسلة "هارولد" في تعليمات البرمجة ١٧-١.

http://www.theharolds.com/genealogy.xml#string(1,"Harold")

ويقوم ذلك باستهداف الموقع الذي يسبق مباشرة حرف M في هارولد في العنصر Name الخاص بتشارلز ولتر هارولدز، ولا يماثل الإشارة إلى العنصر الكامل Name، الذي سوف يفعله المحدد المبني على العنصر.

يمكنك إضافة وسيط الموضع الثالث الاختياري لتحديدكم عدد الأحرف التي سوف يتم استخدامها في الحق الخاص بالبداية الخاصة بمماثلة السلسلة، فعلى سبيل المثال، تقوم هذه الأهداف أياً كانت بالمتابعة المباشرة للحدث الأول الخاص بالسلسلة "هارولد" لأن هارولد لديه ستة حروف.

http://www.theharolds.com/genealogy.xml#string(1,"harold",6)

يقوم الوسيط الرابع الاختياري بتحديد عدد الأحرف التي سوف يتم اختيارها، فعلى سبيل المثال، يقوم URL باختيار الحدث الأول الخاص بالسلسلة الكاملة "هارولد" في تعليمات البرمجة ١٧-١.

http://www.theharolds.com/genealogy.xml#string(1,"Harold", 1.6)

ويتم استخدام السلسلة الفارغة (" ") في سلسلة مصطلح الموقع لتحديد الحروف المعينة في المستند. فعلى سبيل المثال، تقوم URL التالية باستهداف ٢٥٦ حرف في المستند، "ولكي تكون أكثر تحديداً، تقوم URL باستهداف الموضع بين العنصر ٢٥٥ والعنصر ٢٥٦ في المستند".

http://www.theharolds.com/genealogy.xml#string(256, " ")

تم دراسة حالة الأحرف ومسافات الفراغ عند تماثل السلاسل ويتم تجاهل ترميز الأحرف.

بدلاً من طلب أمثلة معينة وتماثل سلسلة معينة يمكنك السؤال عن ذلك باستخدام الكلمة الأساسية all كالوسيط الأول. فعلى سبيل المثال، يقوم القانون باختيار كل الأحداث الخاصة بسلسلة "بيلو" في المستند.

http://www.theharolds.com/genealogy.xml#string(all,"bellau")

وقد ينتج عن ذلك اختيارات غير متجاورة والتي قد لا تفهمها كثير من التطبيقات لذلك يجب أن تستخدم هذه التقنية بحذر.

المصطلح الأصلي للموقع المطلق

يكون المصطلح الأصلي للموقع المطلق هو Origin، ومع ذلك، تكون مفيدة فقط عندما تستخدم مرتبطة مع واحد أو أكثر من مصطلحات الموقع النسبي، في ارتباطات المستندات الداخلية، إلا

وهي، الارتباطات من نقطة واحدة من المستند إلى نقطة أخرى في نفس المستند، ويكون ذلك ضرورياً غالباً للإشارة إلى "العنصر التالي بعد هذا المستند" أو "العنصر الأصلي الخاص بهذا العنصر". يقوم مصطلح الموقع المطلق Origin بالإشارة إلى العنصر الحالي لذلك تكون مثل تلك الإشارات ممكنة.

بدراسة تعليمات البرمجة ١٧-٢ يعرض انزلاق بسيط، في هذا المثال، يشير Origin (). Following (1, SLIDE) إلى الانزلاق التالي في العرض بينما يشير Origin (). Preceding (1, SLIDE) إلى الانزلاق السابق في العرض، ومن المحتمل أن يستخدم ذلك مرتبطاً مع ورقة نمط والتي تقوم بعرض انزلاق واحد في المرة.

تعليمات البرمجة ١٧-٢ : عرض الشرائح

```
<?xml version="1.0"?>
<SLIDESHOW>
 <SLIDE>
  <H1>Welcome to the slide show!</H1>
  <BUTTON xml:link="simple"
       href="origin().following(1,SLIDE)">
   Next
  </BUTTON>
 </SLIDE>

 <SLIDE>
  <H1>This is the second slide</H1>
  <BUTTON xml:link="simple"
       href="origin().preceding(1,SLIDE)">
   Previous
  </BUTTON>
  <BUTTON xml:link="simple"
       href="origin().following(1,SLIDE)">
   Next
  </BUTTON>
 </SLIDE>
 <SLIDE>
  <H1>This is the second slide</H1>
```

```
  <BUTTON xml:link="simple"
        href="origin().preceding(1,SLIDE)">
   Previous
  </BUTTON>
  <BUTTON xml:link="simple"
       href="origin().following(1,SLIDE)">
   Next
  </BUTTON>
 </SLIDE>
 <SLIDE>
  <H1>This is the third slide</H1>
  <BUTTON xml:link="simple"
        href="origin().preceding(1,SLIDE)">
   Previous
  </BUTTON>
  <BUTTON xml:link="simple"
        href="origin().following(1,SLIDE)">
   Next
  </BUTTON>
 </SLIDE>
 ...
 <SLIDE>
  <H1>This is the last slide</H1>
  <BUTTON xml:link="simple"
        href="origin().preceding(1,SLIDE)">
   Previous
  </BUTTON>
 </SLIDE>

</SLIDESHOW>
```

وعامةً، يتم استخدام مصطلح الموقع Origin فقط في URL النسبي الكلي في مؤشـــرات X، وإذا تم اشتمال أي جزء من URL يجب أن يكون مماثلاً مع URL الخاص بالمستند الحالي.

قياس نطاق النص

قد يكون من المهم في بعض التطبيقات تحديد نطاق للنص أكثر من نقطة معينة في المستند، ويمكن إتمام ذلك بواسطة القياسات، ويبدأ القياس عند مؤشر من مؤشرات X ويستمر في مؤشر آخر من مؤشرات X.

وتتم الإشارة إلى القياس بواسطة الكلمة الأساسية () Span ويتم استخدامها كمصطلح موقع، ومع ذلك، يكون الوسيط الخاص بالمصطلح () Span عبارة عن مصطلحين للموقع ويتم فصلهم بواسطة فصلة تشير إلى البداية والنهاية الخاصة بالقياس، وإذا كان ذلك مصطلحات الموقع النسبي، يكون بذلك المصطلح الذي يسبق القياس هو المصدر لكل من المصطلحين.

فعلى سبيل المثال، افترض أنك تريد اختيار كل شيء بين العنصر الأول Person والعنصر الأخير Person في genealogy. xml وتقوم مؤشرات X بإتمام ذلك.

```
root().span(child(1,PERSON),child(-1,PERSON))
```

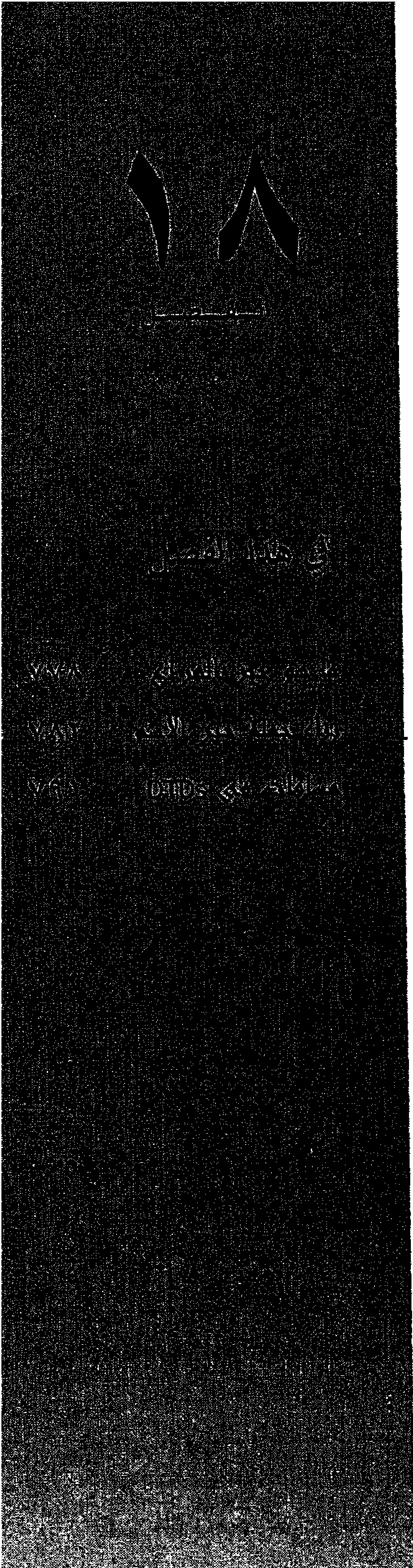
خلاصة

لقد تعلمت في هذا الفصل عن مؤشرات X، وعلى وجه الخصوص تعلمت الآتي:

- تشير مؤشرات X إلى أجزاء معينة خاصة بالمواقع في مستندات XML.
- يشير مصطلح الموقع المطلق id إلى العنصر المحتوي على القيمة المعينة الخاصة بسمة النوع ID.
- يشير مصطلح الموقع المطلق root إلى العنصر الجذر الخاص بمستندات XML.
- يشير مصطلح الموقع المطلق html إلى الاسم المعتمد الخاص بمستند HTML.
- يمكن أن يتم سلسلة مصطلح الموقع النسبي لعمل محددات مركبة أكثر تعقيداً. ويسمى المصطلح الذي يكون نسبياً بالنسبة للمصطلح مصدر الموقع.
- ويشير مصطلح الموقع النسبي Child إلى التابع المباشر الخاص بمصدر الموقع.
- يشير مصطلح الموقع النسبي descendant إلى أي عنصر يتم احتوائه في مصدر الموقع.
- يشير مصطلح الموقع النسبي ancestor إلى العنصر الذي يحتوي مصدر الموقع.
- يشير مصطلح الموقع النسبي Preceding إلى أي عنصر يأتي بعد مصدر الموقع.
- يشير مصطلح الموقع النسبي Following إلى أي عنصر يلي مصدر الموقع.

- يقوم مصطلح الموقع النسبي Psibling بالاختيار من العناصر المتقاربة التي تسبق العنصر المستهدف.
- يقوم مصطلح الموقع النسبي Fsibling بالاختيار من العناصر التي تلي العنصر المستهدف.
- يحتوي كل مصطلح موقع نسبي ما بين واحد وأربعة وسطاء وهي العدد C والنوع C وأسم السمة وقيمة السمة.
- ويكون الوسيط الأول الخاص بمصطلح الموقع النسبي هو عدد يحدد الموضع النسبي الخاص بالعقدة المستهدفة أو الكلمة الأساسية all.
- ويكون الوسيط الثاني بالنسبة لمصطلح الموقع النسبي يحدد النوع الخاص بالعقدة المستهدفة وقد يكون الاسم الخاص بالعنصر أو أحد مفاتيح الكلام. #element و#pi و#comment و#text و#cdata و #all.
- يقوم الوسيط الثالث بالنسبة لمصطلح الموقع النسبي بتحديد الاسم الخاص بالسمة المملوكة بواسطة العقدة المستهدفة.
- ويقوم الوسيط الرابع بالنسبة لمصطلح الموقع النسبي بتحديد القيمة الخاصة بالسمة الخاصة بالعقدة المستهدفة.
- ويشير مصطلح الموقع String إلى كتلة معينة من النص في مصدر الموقع.
- يشير القياس إلى النطاق الخاص بالمستند بدلاً من عنصر واحد معين فقط.

يقوم الفصل القادم باستكشاف حيز الفراغ، ويستخدم Namespace كوسيلة لاستخراج العنصر من المستند الذي يتم تكوينه في تطبيقات XML المتعددة، فعلى سبيل المثال، يسمح حيز الفراغ لك بالاستخدام الفوري لمصطلحين مختلفين للمصطلح XML والذي يعرف نفس العناصر بطريقة غير متوافقة.



Namespace

قد تكون لغة XML مفيدة في كتابة المستندات التي تستخدم مصطلح ترميزي مفرد "راجع مثـــال البيسبول الخاص بالفصلين ٥،٤" وتكون أكثر نفعاً في خلط ومماثلة الرموز الخاصـــة بتطبيقـــات XML المختلفة، لذلك لا تكون لغة XML معزولة، فعلى سبيل المثال، قد تريد اشـــتمال العنصـــر Biography داخل كل عنصر من العناصر الخاصة بـــالعنصر Player، ولأن الســـيرة الذاتيـــة تحتوي أساساً على تكوين حر ونصف منسق، ويكون ذلك ملائماً لكتابتها في نموذج جيد خـــاص بلغة XML بدون إعادة دعوة كل العلامات الخاصة بالفقرات، وفواصل الأسطر وقائمـــة البنـــود والعناصر المكتوبة بخط أسود عريض، وهكذا من البداية.

وتكون المشكلة، في ناحية أخرى، عند خلط ومماثلة العلامـــــات الخاصـــة بتطبيقـــات XML المختلفة، فإنه من المحتمل أن تجد نفس العلامة يتم استخدامها من أجل شيئين مختلفيـــــن، فـــهل يكون Title عنوان لصفة أو عنوان لكتاب؟ أو يكون Address عنوان البريد الخاص بشـــركة أو عنوان البريد الإلكتروني الخاص بويب الرئيسي؟ ويقوم حيز الاسم بإزالة الالتباس الخاص بـــهذه الأمثلة بواسطة ارتباط URL مع كل مجموعة العلامة وتعلق البادئة بكل عنصر للإشارة إلـــى أي مجموعة تكون منتمية، وهكذا، يكون لديك كلاً من العنـــاصر Book Title و title : Html أو العناصر Postal :Address و HTML :Address بدلاً من نوع واحد خاص بـــالمصطلح Title أو Address. ويقوم هذا الفصل بعرض كيفية استخدام حيز الفراغ.

ما هو حيز الفراغ

تقوم لغة XML بتمكين المبرمجين بتكوين لغة الترميز الخاصة بهم والخاصة بمشاريعهم الخاصة. ويمكن لهذه اللغة أن تكون مشتركة بين أفراد العمل الذين يعملون في نفس المشروع فـــــي كـــل أنحاء العالم. ويكون XSL هو مثال واحد معين خاص بذلك. وتكون XSL هي نفسها التطبيقـــــات XML الخاصة بمستندات XML النمطية.

يجب أن تكون لغة التحويل الخاصة بالمصطلح XSL، عشـــوائية المخرجـــات، XML جيـــدة النماذج وإمكانية اشتمالها على XSL نفسها. وهكذا، فإنك تحتاج إلى وسيلة واضحة للتعريف بيـــن عناصر XML وهي تعليمات التحويل الخاصة بالمصطلح XML وعناصر إخراج XML، حتى لـــو كانوا يحملون نفس الاسم.

ويكون حيز الأسماء هو الحل، كما يقوموا بالسماح لكل عنصر وسمة في المستند بـــأن يتـــم وضعهم في حيز أسماء مختلف، ويتم وضع عناصر XML والتي تتضمن تعليمات التحويل فـــي حيز الاسم. http: //www.w3.org/xsl/transform/1.0 ويمكن إعادة وضع عنـــاصر XML والتي تكون جـــزءاً مـــن المخرجـــات فـــي بعـــض حـــيز الاســـم الملائمـــة الأخـــرى مثـــل http://www.w3.org/tr/rec.html4.0 أو http://www.w3.org/xsl/format/1.0. ولا يكون هناك أهمية لحيز الأسماء طالما كانت مختلفة.



وإذا كنت معتادا مع مفهوم حيز الاسم كما تم استخدامهم في C++ وبرامج اللغة الأخرى، يجب أن تضع مفاهيمك السابقة جانباً قبل قراءة الإضافات، ويكون حيز أسماء XML مشابهاً لذلك، ولكن ليس نفس الشيء كما في حيز الأسماء المستخدم في البرمجة، وعملياً، ليس من الضروري أن يكون حيز الاسم مجموعة بدون تكرار.

في القائمة ١٥-٢ يظهر التحويل من مصطلح المصدر إلى كائنات تنسيق XSL مبدئياً في فصل ١٥، ألا وهو انظر الفصل ١٥. كما تقوم بتنظيم ورقة نمط XSL والتي يتم تحويلها من مدخلات XML إلى كائنات تنسيق XML، ويقوم برامج التنسيق بالتفريق بين العناصر التي هي تعليمات XSL والبيانات العرفية الخاصة بالمخرجات بواسطة استخدام حيز الاسم. ويقوم العنصر الخاص بحيز الاسم http://www.w3.org/xsl/trancform/1.0 بتمثيل التعليمات المحولة ويقوم أي عنصر في حيز الاسم http://www.w3.org/xsl/format/1.0 باحتواء جزء خاص بالمخرجات.

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
 xmlns:fo="http://www.w3.org/XSL/Format/1.0"
 result-ns="fo" indent-result="yes">

 <xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">

   <fo:layout-master-set>
    <fo:simple-page-master page-master-name="only">
     <fo:region-body/>
    </fo:simple-page-master>
   </fo:layout-master-set>

   <fo:page-sequence>

    <fo:sequence-specification>
     <fo:sequence-specifier-single page-master-name="only"/>
```

```
</fo:sequence-specification>

 <fo:flow>
   <xsl:apply-templates select="//ATOM"/>
 </fo:flow>

</fo:page-sequence>

</fo:root>
</xsl:template>

<xsl:template match="ATOM">
  <fo:block font-size="20pt" font-family="serif">
    <xsl:value-of select="NAME"/>
  </fo:block>
</xsl:template>

</xsl:stylesheet>
```

وأكثر تحديداً، يوجد هذا العنصر في حيز الاسم http://www.w3.org/xsl/format/1.0

- stylesheet
- template
- apply-templates
- value-of

وتكون هذه العناصر الموجودة في حيز الأسماء http://www.w3.org/xsl/format/1.0 كائنات تنسيق وإجراء خاصة بالمخرجات.

- root
- layout-master-set
- simple-page-master
- region-body
- sequence-specification
- sequence-specifier-single

- page-sequence
- block

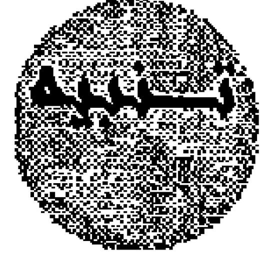
تحتوي العناصر مع بادئة XSL الاسم المؤهل مبتدئاً بالبادئة:

- xsl:stylesheet
- xsl:template
- xsl:apply-templates
- xsl:value-of

ومع ذلك، تقوم الأسماء باستخدام URL أفضل في البادئة:

- http://www.w3.org/XSL/Transform/1.0:stylesheet
- http://www.w3.org/XSL/Transform/1.0:template
- http://www.w3.org/XSL/Transform/1.0:apply-templates
- http://www.w3.org/XSL/Transform/1.0:value-of

وجوهرياً، يمثل الاسم المؤهل القصير اللقب الذي يتم استخدامه فقط داخل المستند لأن URLs تحتوي عادتاً على حروف مثل % و / والتي تكون غير شرعية بالنسبة لأسماء XML، ومع ذلك، تقوم الأسماء المؤهلة بجعل المستند أسهل قليلاً في الكتابة والقراءة.



يكون حيز الاسم من XML هو التوصية المكتبية W3C، ويقوم W3C باعتبارها كاملة، بعيداً عن الأخطاء الصغيرة المحتملة والشرح. وبالرغم من كل تحديدات XML من W3C ذلك أكثرهم جدلاً.

ويشعر كثير من الناس بقوة جداً أن هذا المستوى يحتوي على أخطاء جوهرية، وتدور مناقشات الاعتراض حول أن حيز الأسماء يكون، في المحاربة، وغير متوافق مع DTD والصلاحية، وحيث أنه ليس لدى رأي قوي في هذه الطريقة أو الأخرى الإجماع على شيء ويكون حيز الأسماء جزء شائك بالنسبة لكثير من المواصفات المرتبطة بلغة XML مثل XSL و XHTML لذلك يجب أن تفهمهم، وعلى الرغم من ذلك، يقوم الكثير من الناس المبرمجين والمؤلفين باختيار تجاهل هذه المواصفات الخاصة بعملهم.

بناء جملة حيز الاسم

لقد تم وضع حيز الاسم في طبقة عليا بمواصفات XML 1.0، يقوم معالج XML 1.0 والـــذي لا يعرف شيئاً عن حيز الاسم بقراءة المستند الذي يستخدم حيز الاسم دون وجود أي أخطـــاء، ولا تقوم المستندات التي تستخدم حيز الاسم بفصل XML الموجودة (على الأقل التي لم يتم فحصـــها للصلاحية)، ولا يجب على المستخدمين انتظار برامج الشركات سيئة الســـمة غـــير المنضبطـــة للحديثات غلاية الثمن قبل استخدام حيز الاسم.

تعريفات حيز الاسم

يتم تعريف حيز الاسم باستخدام السمة XM 1ns: Prefix بتطبيق العناصر القابلة للتطبيـــق، ثـــم إعادة إحلال Prefix بواسطة البادئة الفعلي المستخدم في حيز الاسم، وتكـــون القيمـــة الخاصـــة بالسمة هي URL الخاصة بحيز الاسم، فعلى سبيل المثال، تقوم هذه العلامة XS1: Style sheet بمرافقة البادئة XS1 مع URL http://www.w3.org/xsl/transform/1.0

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
```

يمكن للبادئة XS1 بعد ذلك أن يكون متعلقاً بالعنصر المحلي وأسماء السمات خلال العنصـــر Xs1: Style/esheet لتحديدهـــم كمنتميـــن فصـــل لحـــيز الاســـم URL http://www.w3.org/xsl/transform/1.0 ويتم فصل البادئة عن الاسم المحلـــي بواســـطة علامة النقطتين. وفي القائمة ٢-١٤، وهي ورقة نمط XSL الأساسية الخاصة بالجدول الـــدوري والتي تم عرضها أولاً في الفصل ١٤. ويوضح بواسطة استخدام البادئة XS1 في العناصر Style Sheet, Templates و apply – templates.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="PERIODIC_TABLE">
    <html>
      <xsl:apply-templates/>
    </html>
  </xsl:template>

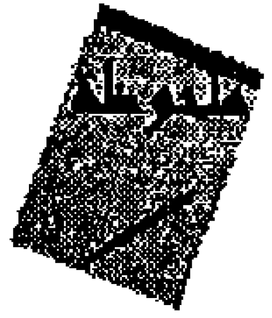
  <xsl:template match="ATOM">
    <P>
      <xsl:apply-templates/>
```

```
</P>
</xsl:template>

</xsl:stylesheet>
```

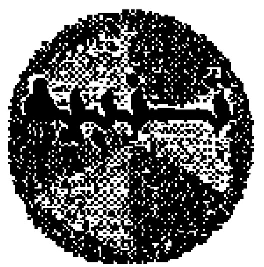
ويكون URL الذي يعرف حيز الفراغ رسمي، ويكون هدفها الرئيسي هو جمع وإزالة اللبس في العناصر وأسماء السمات في المستند، وليس من الضروري الإشارة إلى شيء، وبالتحديد، ولا يوجد ضمان بأن المستند في URL، سوف يقوم بوصف بناء الجملة المستخدم في المستند، أو في حالة وجود أي مستند في URL، ويجب ذكر ذلك، إذا كانت XML قانونية بالنسبة لتطبيقات XML المعينة، وبذلك تكون URL اختيار جيد بالنسبة لتعريفات حيز الاسم.

يمكن لبادئة حيز الاسم أن يكون أي اسم XML قانوني والذي لا يحتوي على علامتين، إعادة النداء في الفصل ٦. بأن الاسم القانوني XML يجب أن يبدأ بحرف أو تسطير أسفل السطر (–) وقد تحتوي الأحرف التالية في الاسم على حروف وأرقام وتسطير أسفل السطر والوصلة والنقط وقد لا يحتوي على مسافات فارغة.



يوجد بادئتين غير مسموح لهم وهم xml و xmlns، ويتم تعريف البادئة XML للإشارة إلى http://www.w3.org/xml/1998/name ، يعتاد البادئة XMLN على ربط العناصر بغير الاسم وهي بذلك تكون متاحة مثل البادئة ليرتبط بها.

والأكثر من عدم السماح لحروف علامة النقطتين في أسماء XML "بعيداً عن استخدامها في البادئات المنفصلة والأسماء المحلية". لا يحتوي حيز الأسماء على تأثير مباشر على بناء جملة XML القياسي. يجب أن يكون المستند الذي يستخدم حيز الاسم نموذج جيد عندما تتم قرأته بواسطة معالج لا يعرف شيئاً عن حيز الاسم. وإذا كان المستند صالحاً، فإنه يجب أن يكون صالحاً بدون تحديد اعتباري لحيز الاسماء، وبالنسبة لمعالج XML، فإن المستند الذي يستخدم حيز الأسماء يكون فقط مستند ذا منظر مضحك والتي قد تحتوي فيها بعض العناصر وأسماء السمات على علامة النقطتين.



يقوم حيز الأسماء بتقديم المشاكل الخاصة بالصلاحية، فإذا تمت كتابة DTD بدون بادئة حيز الأسماء قبل أن يمكن استخدامها في صلاحية المستندات التي تستخدم البادئات. فعلى سبيل المثال، أدرس إعلان هذه العناصر:

```
<!ELEMENT DIVISION (DIVISION_NAME, TEAM+)>
```

ويجب أن تتم إعادة كتابتها مثل ذلك إذا تم إعطاء على العناصر بادئة حيز الاسم bb:

```
<!ELEMENT bb:DIVISION (bb:DIVISION_NAME, bb:TEAM+)>
```

ويعني ذك إنه لا يمكن استخدام نفس المصطلح DD لكل من المستندات التي تحتوي على حيز الاسم والتي لا تحتوي على حيز الاسم حتى إذا كانوا يستخدموا نفس المصطلحات. وفي الواقـــع، لا يمكن استخدام نفس مصطلح DTD مع المستندات التي تستخدم نفس مجموعات العلامات وحيز الأسماء، ولكن البادئات المختلفة لأن DTDs تكون مربوطة مع البادئات الواقعية أكثر من URLs الخاصة بحيز الاسم.

حيز الاسم المتعدد

في القائمة ١٤-٢ لا يتم في الحقيقة وضع عناصر HTML في حيز الاسم، ولكـــن ليـــس مـــن الصعب عمل ذلك، ويتم توضيح ذلك في القائمة ١-١٨، ويكون html هو البادئة الاصطلاحـــي الخاص بعناصر HTML تماماً، سيكون XSL هو البادئة الاصطلاحي الخاص بـــالمصطلح XSL، وفي هذا المثال، يقوم العنصر xsl:style/sheet بإعلان حيزي أسماء مختلفين، أحدهم خـــاص بالمصطلح XML والأمر خاص باللغة HTML.

تعليمات البرمجة ١٨-١: تقوم ورقة النمط XML باستخدام حيز الاسم http://www.w3.org/tr/rechtml4.0 الخاص بالمخرجات

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
        xmlns:html="http://www.w3.org/TR/REC-html40">
<xsl:template match="PERIODIC_TABLE">
  <html:html>
    <xsl:apply-templates/>
  </html:html>
 </xsl:template>
<xsl:template match="ATOM">
  <html:p>
    <xsl:apply-templates/>
  </html:p>
 </xsl:template>
</xsl:stylesheet>
```

ومن المعتاد والأكثر نفعاً وضع السمة XMLNS في عنصر الجذر، فإنه من الممكن أن تظهر في مستند الاسم فقط داخل العنصر الذي يتم الإعلان فيه. تابع القائمة ٢-١٨، ويكون البادئة html قانوني في حالة وجوده فقط في العنصر xsl: template والذي تم إعلانه فيه، ولا يمكن أن يتم تطبيقها في روافد أخرى من القوانين، إلا إذا أعلنوا بصفة منفصلة حيز الاسم html.

تعليمات البرمجة ١٨-٢: يتم إعلان ورقة النمط XSL مع حيز الاسم http://www.w3.org/tr/rec-htm/4.0 في روافد القوانين

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="PERIODIC_TABLE"
          xmlns:html="http://www.w3.org/TR/REC-html40">
    <html:html>
      <xsl:apply-templates/>
    </html:html>
  </xsl:template>

  <xsl:template match="ATOM">
    <p>
      <xsl:apply-templates/>
    <p>
  </xsl:template>

</xsl:stylesheet>
```

يمكن إعادة تعريف حيز الاسم في العنصر التابع، فعلى سبيل المثال، تابع ورقة النمط XSL في القائمة ١٨-٣، هنا يظهر البادئة XSL في عناصر مختلفة ليشير إلى http://www.w3.org/xsl/transform/1.0 و http://www.w3.org/xsl/format/1.0 بالتناوب. وعلى الرغم من أن كل عنصر يحتوي على البادئة XSL، فغن مصطلحات XSL المحولة وتنسيق كائن XSL يبقى ساكناً في حوار الأسماء المختلفة لأن معنى البادئة XSL يتغير من عنصر لآخر.

تعليمات البرمجة ١٨–٣: إعادة تعريف البادئة XSL

```
<?xml version="1.0"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

  <xsl:template match="/">
   <xsl:root xmlns:xsl="http://www.w3.org/XSL/Format/1.0">

     <xsl:layout-master-set>
       <xsl:simple-page-master page-master-name="only">
         <xsl:region-body/>
       </xsl:simple-page-master>
</xsl:layout-master-set>

     <xsl:page-sequence>

     <xsl:sequence-specification>
      <xsl:sequence-specifier-single page-master-name="only"/>
     </xsl:sequence-specification>

       <xsl:flow>
         <xsl:apply-templates select="//ATOM"/
           xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"/>
       </xsl:flow>

     </xsl:page-sequence>

    </xsl:root>
  </xsl:template>
<xsl:template match="ATOM">
    <xsl:block font-size="20pt" font-family="serif"
      xmlns:xsl="http://www.w3.org/XSL/Format/1.0">
      <xsl:value-of select="NAME"
       xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"/>
    </xsl:block>
  </xsl:template>
</xsl:stylesheet>
```

ومع ذلك، يكون ذلك إضراب غير ضروري، وننصح بشدة تجنب ذلك، يوجـــد كثـــير مـــن البادئات التي يمكن أن تعمل، لذلك غالباً لا توجد حاجة لإعادة استخدام البادئـــات داخـــل نفـــس المستند. وتكون الأهمية الرئيسية لذلك، في إنه إذا حدث وقام مستندان مختلفان لمؤلفين مختلـــف بإعادة استخدام شقي البادئة فأنهم يصبحوا متحدين، ويكون ذلك ســـبب جيـــد لتجنـــب البادئـــات القصيرة مثل m و a و x والذين من المحتمل أن يتم إعادة استخدامهم لأغراض مختلفة.

السمات

منذ انتماء السمات إلى عناصر معينة، ويكونوا غير ملتبسين بسهولة أكثر من أســـماء الســـمات المتشابه بدون جواز الأسماء، وتدريباً، وليس من المبكر حيز الاسم إلـــى الســـمات كمـــا فـــي العناصر. فعلى سبيل المثال و ٢١ من أبريل ١٩٩٩، العمل التمـــهيدي الخـــاص بالمواصفـــات المطلوبـــة وهـــو انـــه يجـــب أن تقـــع كـــل عنـــاصر تحويـــل XSL فـــي حـــيز الاســـم http://www.w3.org/XSL/Transform/1.0 ومع ذلك، فـــهذا لا يتطلـــب أن تكـــون السمات الخاصة بهذه العناصر موجودة في أي حيز اسم معين "وفي الواقع، فإنـــه يتطلـــب أن لا يتواجد في أي حيز اسم" ومع ذلك، يمكن أن تتعلق ببادئة حيز الاسم الخاص بالســـمات إذا كـــان ضرورياً، فعلى سبيل المثال، يكون العنصر Player وكل سماته موجـــودة فـــي حـــيز الاســـم http://meta/ab.unc.edulxml/basball

```
<bb:PLAYER xmlns:bb="http://metalab.unc.edu/xml/baseball"
 bb:GIVEN_NAME="Tom" bb:SURNAME="Glavine"
 bb:POSITION="Starting Pitcher" bb:GAMES="33"
 bb:GAMES_STARTED="33" bb:WINS="20" bb:LOSSES="6"
bb:SAVES="6"
 bb:COMPLETE_GAMES="4" bb:SHUT_OUTS="3" bb:ERA="2.47"
 bb:INNINGS="229.1" bb:HOME_RUNS_AGAINST="13"
 bb:RUNS_AGAINST="67" bb:EARNED_RUNS="63" bb:HIT_BATTER="2"
 bb:WILD_PITCHES="3" bb:BALK="0" bb:WALKED_BATTER="74"
 bb:STRUCK_OUT_BATTER="157"/>
```

وقد يكون ذلك مفيد بالمناسبة إذا كانت هناك رغبة في اتحاد سمات خاصة باثنين من تطبيقات XML المختلفة في نفس المستند.

ومن الممكن "على الرغم من أنه غالباً لا يشير إلى شيء" ربط نفس حيز الاســـم URL مـــع بادئين أثنين مختلفين، ولا يوجد سبب حقيقي لفعل ذلك، والسبب الوحيد الذي ظهر ببساطة هو من اجل التحذير بأن حيز الاسم الكامل الخاص بالسمات والتي يجـــب أن ترضـــي بقوانيـــن XML's الخاصة بالعنصر الذي لا يحتوي على أكثر من سمة واحدة مع نفس الاسم، فعلى سبيل المثـــال، يكون ذلك غير قانوني لأن baseball: given, name bb: Given-name يكونوا متشابهان.

```
<bb:PLAYER xmlns:bb="http://metalab.unc.edu/xml"
        xmlns:baseball="http://metalab.unc.edu/xml"
  bb:GIVEN_NAME="hank" bb:SURNAME="aaron"
  baseball:GIVEN_NAME="Henry" />
```

ومن ناحية أخرى، فإن URL لم تقوم واقعياً بالفحص لمعرفة ما الذي تشير إليه، ويقوم http://metalab.ure.edu/xml و http://www.metalab.ure.edu/xml بالإشارة إلى نفس الصفحة. ومع ذلك يكون هذا واقعياً:

```
<bb:PLAYER xmlns:bb="http://metalab.unc.edu/xml"
        xmlns:baseball="http://www.metalab.unc.edu/xml"
  bb:GIVEN_NAME="Hank" bb:SURNAME="aaron"
  baseball:GIVEN_NAME="henry" />
```

حيز الاسم الافتراضي

في المستندات الطويلة والتي تحتوي على كثير من الترميزات والتي تكون جميعها من نفس الاسم، فإنه من الممكن إيجاد أنه في الملائم إضافتها بادئة لكل اسم عنصر، كما يمكن أن تربط حيز الاسم الافتراضي مع العنصر ويقوم العنصر التابع الخاص به باستخدام السمة xmlns مع البادئة، ويتم اعتبار العنصر نفسه، مثله مثل كل البادئات الخاصة به موجودة في تعريف حيز الاسم إلا إذا كانوا يحتون على بادئة واضح، فعلى سبيل المثال، كل من القائمة ٤-١٨ ورقة نمط XSL والتي لا تبدأ بعناصر التحويل XSL مع XSL كالمعتاد.



لا توجد السمات في حيز الأسماء الافتراضي، بل يجب أن يكونوا بادئات واضحة.

تعليمات البرمجة ١٨-٤: تقوم ورقة النمط XSL باستخدام حيز الاسم الافتراضي

```
<?xml version="1.0"?>
<stylesheet
 xmlns="http://www.w3.org/XSL/Transform/1.0"
 xmlns:fo="http://www.w3.org/XSL/Format/1.0"
 result-ns="fo">

 <template match="/">
```

```
<fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">

  <fo:layout-master-set>
    <fo:simple-page-master page-master-name="only">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence>

   <fo:sequence-specification>
    <fo:sequence-specifier-single page-master-name="only"/>
   </fo:sequence-specification>

    <fo:flow>
      <apply-templates select="//ATOM"/>
    </fo:flow>

  </fo:page-sequence>

 </fo:root>
</template>

<template match="ATOM">
  <fo:block font-size="20pt" font-family="serif">
    <value-of select="NAME"/>
  </fo:block>
</template>

</stylesheet>
```

وقد يكون أحسن استخدام لحيز الاسم الافتراضي هو تعلق حيز الاسم بكل عنصر في المستند الموجود والتي سوف يتم إضافة العلامات إليها في لغات مختلفة، فعلى سبيل المثال، إذا قمت بوضع Mathml في مستند HTML فإنه يجب أن تضيف فقط البادئات الخاصة بعناصر Mathml كما يمكن وضع كل عناصر HTML في حيز الاسم http://www.w3.org/tr/rec/html/4.0 ببساطة بواسطة إعادة وضع علامة البداية <html> مع هذه العلامة:

```
<html xmlns="http://www.w3.org/TR/REC-html40">
```

ولا تحتاج إلى إضافة بقية الملفات، لأن علامات Mathml لا زالت تحتاج إلى أن تكون في حيز اسم منفصل، ومع ذلك، كلما طالت الفترة التي لا يتم فيها الخلط مع الكثير من ترميزات HTML، يمكن ببساطة إعلان السمة Xmlns في عنصر الجذر الخاص بالمصطلح Mathml. ويقوم ذلك بتعريف حيز الاسم لعناصر Mathml والتي تتجاوز حيز الاسم الافتراضي الخاص بالمستند الذي يكمل Mathml وتقوم القائمة ١٨-٥ بتوضيح:

تعليمات البرمجة ١٨-٥: يتم تضمين عنصر الرياضيات Mathml في النموذج الجيد الخاص بمستند HTML والتي تستخدم حيز الاسم

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/TR/REC-html40">
  <head>
    <title>Fiat Lux</title>
    <meta name="GENERATOR" content="amaya V1.3b" />
  </head>
  <body>

    <P>And God said,</P>

    <math xmlns="http://www.w3.org/TR/REC-MathML/">
      <mrow>
        <msub>
          <mi>&#x3B4;</mi>
          <mi>&#x3B1;</mi>
        </msub>
        <msup>
          <mi>F</mi>
          <mi>&#x3B1;&#x3B2;</mi>
        </msup>
        <mi></mi>
        <mo>=</mo>
        <mi></mi>
        <mfrac>
          <mrow>
            <mn>4</mn>
```

```
            <mi>&#x3C0;</mi>
          </mrow>
          <mi>c</mi>
        </mfrac>
        <mi></mi>
        <msup>
          <mi>J</mi>
          <mrow>
            <mi>&#x3B2;</mi>
            <mo></mo>
          </mrow>
        </msup>
</mrow>
    </math>

    <P>and there was light</P>

  </body>
</html>
```

توجد كل من Math وMrow وMsub وMfracr mi cmo وmn وMsup في حيز الاسم http://www.w3. org/TR/REC-MathML/، حتى إذا كان المستند يحتويهم يستخدم حيز الاسم http://www.w3.org/tr/rec-html 4.0

مسافات في DTDs

لا تجعل خواص الأسماء على أي إعفاء خاص من القوانين الطبيعية القادمة بالنماذج الجيدة والصلاحية. وبالنسبة للمستند الذي يستخدم حيز الاسم ليصبح صالحاً، لذا يجب أن تعلن السمة xmlns في الخاص بهذه العناصر، والتي يتعلقون بها، والأكثر من ذلك، يجب أن يتم إعلان العناصر والسمات باستخدام البادئات التي يستخدموها في المستند، فعلى سبيل المثال، إذا كان المستند يستخدم العنصر Math: Subset، وبعد ذلك يجب أن تقوم DTD بإعلان العنصر Math: Subset وليس العنصر الكامل Sabset "ومن المسلم به إنه لا يتم تطبيق هذه القوانين على مستندات النماذج البعيدة والتي تناقش ذلك بعيداً" فعلى سبيل المثال:

```
<!ELEMENT math:subset EMPTY>
```

يمكن لقيمة العنصر الافتراضي والسمات Implied# أن تساعد هنا، فعلى سبيل المثال، يقوم هذا الإعلان Attlist بوضع كل عنصر Math:Subset في حيز الاسم

إلا إذا لم تكن محددة في المستند. http://www.w3.org/tr/rec-mathtml

<!ATTLIST math:subset

عند العمل مع المستندات الصالحة، تقوم الأسماء الافتراضية بصفة خاصة بإثبات نفع خـاص عندما لا تتطلب إضافة بادئات إلى العناصر وبإضافة بادئات إلى العناصر فـي تطبيقـات XML والتي لا تستخدمها DTD سوف يؤدي ذلك إلى وقف الصلاحية.

ومع ذلك، يوجد حدود واضحة لمعرفة إلى أي مدى سوف تأخذنا جواز الأسماء الافتراضيـة، وعلى وجه الخصوص، فأنهم لا يكونوا كافيين للتفرقة بين عنصرين يسـتخدموا اسـم العنصـر بطريقة غير متوافقة. فعلى سبيل المثال، إذا قام DTD بتعريف HEAD كمكمل للعنـاصر Meta وTitle وتقوم بعض DTD بتعريف Head كمكمل PCDATA#، وبعـد ذلـك جـب أن تقـوم باستخدام البادئات في DTD والمستند للتفرقة بين العنصرين HEAD المختلفين.

يوجد مجهودين مختلفين لتطوير أثناء العملية والتي قد تحل "وقد لا تحل" المشـكلة الخاصـة بدمج DTDs غير المتوافق في مجالات مختلفة، وقد تقوم طريقة XML بإمداد اسـتبدالات أكـثر نشاطاً بالنسبة DTDs، وقد تمكن إجراء XML المستندات المختلفة تشترك مع اختلافات أكثر بيـن أي الأجزاء جانت فرايت. ومع ذلك، لم يتم إغلاق أي منها لانتهائه، وبناءاً على ذلك، حتى الآن، دمج DTDs غير المتوافقة، سوف يؤدي إلى المطالبة بإعادة كتابة DTD والمستندات لاسـتخدام البادئات.



إذا كان لديك سؤال عن ما إذا كانت المستندات التي تستخدم حيز الاسـم نمـاذج جيدة أو صالحة، فيجب أن ننسى أي شيء عرفته عن حيز الاسم، وتتم معاملـة المستندات ببساطة مثل أي مستند عادي من مستندات XNL والـذي قـد يحـدث ويحتوي على بعض العناصر وأسماء السمات التي تحتوي على الأعمدة. وتكـون المستندات نماذج جيدة وصالحة كما هي عند عدم اعتبار حيز الاسم

خلاصة

يقوم هذا الفصل بتوضيح كيفية التعامل مع حيز الاسم، وعلى وجه الخصوص قد تعلمت:

- يقوم حيز الاسم بالتعريف بين العناصر والسمات الخاصة بنفس الاسم من تطبيقات XML المختلفة.
- يتم إعلان حيز الاسم بواسطة السمة xmlns والتي تكون قيمتها هي URL الخاصة بحيز الاسم، وتتم الإشارة إلى المستند بواسطة URL الذي لا يعالج إلى وجوده.

- ويكون البادئة المرتبط بحيز الاسم جزئياً من الاسم الخاص بالسمة xmlns والتي تتبع العمود، فعلى سبيل المثال xmlns: prefix.
- يتم تعلق البادئات مع كل العناصر وأسماء السمات والتي تنتمي لحيز الاسم بواسطة البادئة.
- إذا كانت السمة xmlns لا تحتوي على بادئة، فإنها تقوم بتأسيس حيز اسم افتراضي لهذا العنصر والعناصر التابعة الخاصة بها "ولكن ليس لأي عنصر".
- يجب أن تتم كتابة DTDs بطريقة مثل الطريقة التي يقوم فيها المعالج الذي لا يعرف شيئاً عن حيز الأسماء بإمكانية مزاولة عمله في توزيع وصلاحية المستندات.

وسوف يقوم الفصل القادم بشرح وصف هيكل المصدر وRDF وتطبيقات XML الخاصة بوضع التعليمات البريدية الخاصة بالبيانات التفصيلية وبنية المعلومات.

١٩

هيكل وصف المورد

يُعد نظام وصف المورد The Resource Description Framework (RDF) تطبيق للغة XML لوضع التعليمات البرمجية الخاصة بالبيانات التفصيلية. وهو يناسب وصف مواقع وصفحات الويب وبهذا لا تقوم وسائل البحث بعملية الفهرسة فقط وإنما تقوم أيضاً باستيعاب ما تفهرسه وعندما يصبح RDF ومصطلحاته القياسية سائدة الانتشار على الويب فإنه يمكن إيجاد العناصر المراد البحث عنها يتناول هذا الفصل التعليمات البرمجية التي تخص RDF وذلك عن الموارد والبناء الأساسي وكذلك المختصر الخاص بعملية RDF كما يتناول استخدام الحاويات لجمع قيم الخاصية ومخططات قاعدة البيانات RDF.

ما هو RDF

البيانات التفصيلية هي بيانات عن بيانات ومعلومات عن معلومات، فمثلاً يعد نص الكتاب هو بياناته بينما اسم المؤلف وعنوان الناشر وتاريخ النشر ...الخ هي البيانات التفصيلية. وتوجد للبيانات التفصيلية استخدامات كثيرة على الويب وتتضمن التنظيم والبحث والانتقاء وإضفاء الصفة الشخصية على مواقع الويب. ولا بد للبيانات التفصيلية الدقيقة أن تسهل عملية إيجاد مواقع الويب المرغوب فيها وتجنب المواقع غير المرغوب فيها.

ولكن تكتسب البيانات التفصيلية مثل هذه المميزات يجب أن تتوافق كل من مواقع الويب ووسائل البحث والفهارس على استخدام تنسيق قياسي للبيانات التفصيلية. ونظام وصف المورد هو لغة XML للتطبيق W3C الموصى باستخدامه لوضع التعليمات البرمجية والتبديل وإعادة استخدام البيانات التفصيلية التي تم بناءها. ويمكن لمصطلحات RDF وصف أنظمة المعدل وخرائط الموقع والتفاصيل الخاصة والخدمات العامة وقيود التراخيص وخلاصة.

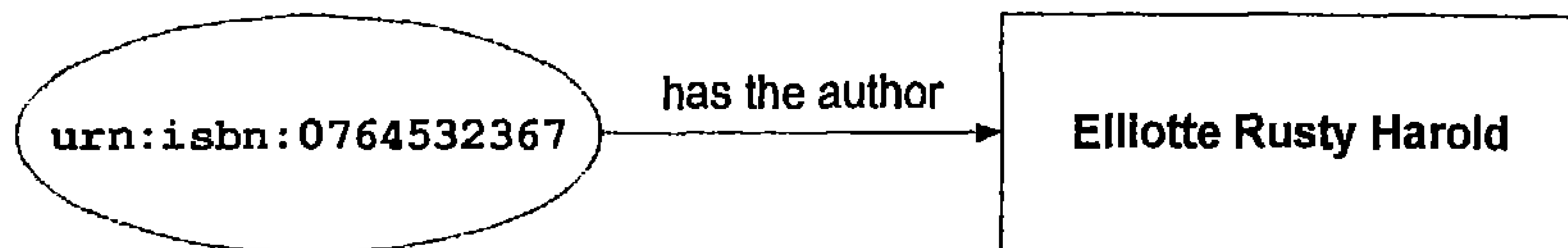
وبوجه عام يجب تعميم مصطلحات البيانات التفصيلية على مستوى الأفراد. من ناحية أخرى يعمل RDF جاهداً على إنشاء نظام عام يحكم كيفية تكوين المعاني والصيغ والبنية الخاصة بالبيانات التفصيلية للمجالات المنفصلة وبهذا يمكن أن تدمج تنسيقات البيانات التفصيلية المتطورة لمجال معين مع تنسيقات متطورة لمجال أخر ومستخدمة في مجال ثالث، دون الخلل بالتعليمات البرمجية الأصلية لقد تم تصميم RDF للتسهيل على برامج الكمبيوتر عند تعاملها مع مواقع الويب يؤدي بدون إلى استكشاف الموارد على المواقع بالإضافة إلى عرض محتوياتها وحساب هذه المحتويات وإظهار مالكها وتحت أي ظروف وتكاليف يمكن استخدامها. كما أنه يوفر إمكانات قد يحتاجها المستخدم الماهر.

تعليمات RDF البرمجية

يقوم مستند أو عنصر RDF بإعداد التعليمات البرمجية عن الموارد. تشير التعليمات البرمجية إلى أن لمورد معين خاصية واحدة أو أكثر. ولكل خاصية نوع "أي اسم واحد" وقيمة واحدة ويمكن أن تكون قيمة هذه الخاصية حرفية مثل سلسلة أو رقم أو تاريخ أو تكون أي شيء آخر.

تتكون التعليمات البرمجية من ثلاثة عناصر: الموارد ونوع الخاصية وقيمة الخاصية. وذلك مثل تعليمات RDF البرمجية التي تشير إلى أن "The book The XML Bible (ISBN: 0-7645-3236-7) has the author Elliotte Rusty Harold" فالمورد هنا هو "The book The XML Bible (ISBN: 0-7645-3236-7)" وخاصية المؤلف لهذا المصدر قيمتها "Elliotte Rusty Harold" والشكل ١٩-١ يوضح الطريقة العامة للوصف المصور لهذه العبارة من تعليمات RDF البرمجية.

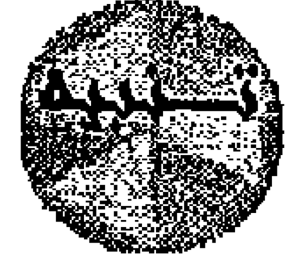
يمكن للمورد أن يأخذ أي شكل يحتوي على.



الشكل ١٩-١ رسم توضيحي لواحدة من تعليمات RDF البرمجية.

يمكن للمورد أن يأخذ أي شكل يحتوي على Uniform Resource Locators (URLs). تعد URI مجموعة هائلة من Uniform Resource Locators (URLs) الأكثر شيوعاً ولكن يمكنهم مطابقة الكتب والعناصر على الصفحة وعروض التليفزيون والأشخاص وخلافه. وفي المثال السابق تستخدم ISBN وكأنها URI للكتاب. وبهذا يمكن أن يكون المورد موقع كامل للويب "http://www.norml.org/" أو أنه صفحة واحدة "http://www.mozilla.org/rdf/doc/index.html" أو عنصر يخص HTML أو XML على صفحة الويب المتطابقة مع XPointer مثل "http://metalab.unc.edu/xml/ mailinglists.html#root().child(1,dt)" أو أنه كتاب "urn:isbn:0764532367" أو شخص "mailto:elharo@metalab.unc.edu" أو أي شيء أخر طالما يمكن بناء URI لها. فالطلب الوحيد للمورد هو URI وفريد ولا ينبغي له أن يكون URI فيمكنه أن يكون أي شيء أخر مثل ISBN.

تقوم الخواص بوصف الموارد حيث توجد لكل خاصية ميزة أو سمة أو علاقات محددة مع المورد. كما لها معنى خاص يمكن التعرف عليه عن طريق اسم الخاصية ومخطط قاعدة البيانات المرتبط بها. ويجب أن يوجد مخطط قاعدة البيانات URI المستخدم في حيز اسم الخاصية. ويعرف مخطط قواعد البيانات القيم أو معدلات القيم التي يسمح بها للخاصية مع أنواع الموارد التي يمكن وصفها.



لا تزال مخططات قواعد البيانات مرحلة التطور فلا داعي للقلق عندما لا تجد مخططاً في مكان يفترض تواجده به. وكذلك يجب ملاحظة أن حيز الاسم RDF وليس متطلب لحيز الأسماء على وجه العموم. وفي الواقع لا تحتاج بعينات حيز الأسماء على الأخص أي من هذه المتطلبات.

تقوم فقط RDF بتعريف صيغ لغة XML بوضع التعليمات البرمجية لثلاثية XML وهي المورد ونوع الخاصية وقيمة الخاصية فلا تقوم بتعريف المصطلحات المستخدمة لوصف الموارد والخواص. وأخيراً يحتاج هذا الطلب أن يكون معنوناً أيضاً وعلى الأقل إذا RDF مستخدم خارج الشبكات المحلية وتكثف الجهود لإخراج مصطلحات قياسية لمعدل المحتوى "PICS 2.0" والمعلومات الشخصية "P3P" وكتالوجات المكتبات الرقمية "Dublin Core" كما يمكن ابتكار مصطلحات أخرى عند الحاجة.

تجمع عبارة تعليمات RDF البرمجية مورد معين مع الخاصية المسماة وقيمتها. تسمى هذه الأجزاء الثلاثة للعبارة المبتدأ والخبر والملحق على التوالي. أي أن المورد هو المبتدأ والخاصية هي الخبر أما قيمة الخاصية فتوصف على أنها الملحق.

وفيما يلي جملة عادية يمكن قراءتها.

Elliotte Rusty Harold is the creator of the Web site at the URL http://metalab.unc.edu/xml/.

ويمكن كتابة هذه الجملة بعدة طرق مثال:

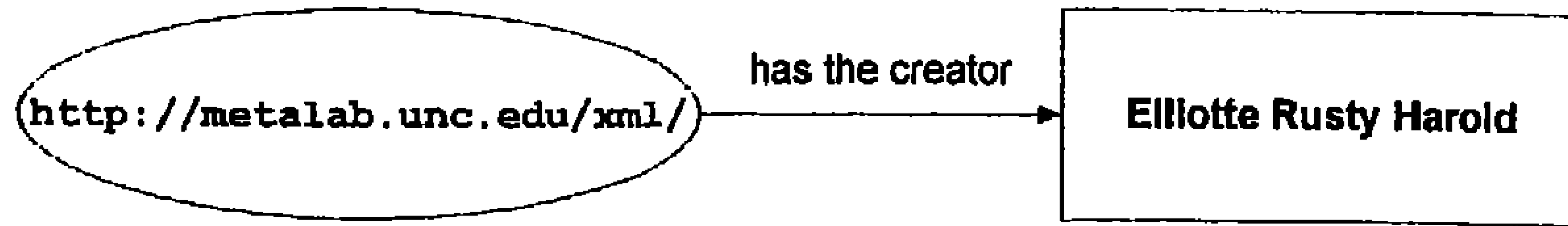
The Web site at the URL http://metalab.unc.edu/xml/ has the creator Elliotte Rusty Harold.

The Web site at the URL http://metalab.unc.edu/xml/ was created by Elliotte Rusty Harold.

The creator of the Web site at the URL http://metalab.unc.edu/xml/ is Elliotte Rusty Harold.

Elliotte Rusty Harold created the Web site at the URL http://metalab.unc.edu/xml/.

وعلى أية حال تعني الصيغ الخمس السابقة نفس الشيء تماماً. ففي كل صيغة يعد المبتدأ هو موقع الويب على URL نحو /http://metalab.unc.edu/xml. والخبر هو خاصية المنشئ والملحق هي قيمة خاصية المنشئ اليبوت روستي هارولد. يوضح "الشكل ١٩-٢" هذه العبارة كما يستوعبها RDF.



الشكل ١٩-٢ رسم توضيحي لواحدة من تعليمات RDF البرمجية.



لا توجد صلة بين المبتدأ أو الخبر الملحق الذي يخص RDF مع هذه المصطلحات في اللغة. ففي الواقع يتمثل جزء من هدف RDF في عزل معنى المبتدأ أو الخبر والملحق لفكرة ما عن دورهم في أي جملة حيث يمكن التعبير عن نفس الفكرة بعدة جمل وفي كل منها يتغير مكان المبتدأ والخبر والملحق.

صيغ RDF الأساسية

تكمن أهمية RDF في أخذ عبارة مفهومة مثل "Elliotte Rusty Harold is the creator of the Web site at the URL http://metalab.unc.edu/xml/" وكتابتها بشكل قياسي منسق على لغة XML والتي بها يمكن لأجهزة الكمبيوتر فهمها.

العنصر الجذري

العنصر الجذري لمستند RDF هو RDF فهذا العنصر بل وصيغ عناصر RDF يتم وضعها الطبيعي في حيز الاسم http://www.w3.org/1999/02/22-rdf-syntax-ns# "العلامة # ليست خطأ مطبعي فهي توجد عندما يرتبط عنصر الاسم مع حيز الاسم وبالتالي الإسفار عن URL صحيح" ويأخذ حيز الاسم إما البادئة rdf أو إنه يجهز مثل حيز الاسم الافتراضي ومثال ذلك عندما تكون البادئة ظاهرة يبدو عنصر RDF الفارغ كما يلي:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!" rdf:Description elements will go here ">
</rdf:RDF>
```

ومع حيز الاسم الافتراضي يبدو هكذا:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!" rdf:Description elements will go here ">
</RDF>
```

عنصر الوصف

تسلسل عبارة RDF في XML كعنصر Description. تعد كل خاصية من المــورد الموصــوف عنصراً تابعاً Description. ويحتوي هذا العنصر التابع هو قيمة الخاصية. ومثال ذلك تحويــل التعليمات البرمجية ١٩-١ عبارة "Elliotte Rusty Harold created the Web site at the URL http://metalab.unc.edu/xml/" إلى RDF.

تعليمات البرمجة ١٩-١: عبارة التعليمات البرمجية في RDF

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description about="http://metalab.unc.edu/xml/">
    <Creator>Elliotte Rusty Harold</Creator>
  </rdf:Description>
</rdf:RDF>
```

ويحتوي عنصر rdf:RDF عبارة برمجية واحدة. وتوضح التعليمات البرمجية لهذه العبـــارة علـــى أنـــها عنصـــر rdf:Description والمـــورد "المبتـــدأ" فـــي هـــــذه العبـــارة هـــو http://metalab.unc.edu/xml/. والخبر هو محتوى عنصر rdf:Description element, <Creator>Elliotte Rusty Harold</Creator> أو الملحق لهذه العبـــارة فـــهو محتـــوى عنصر Creator أي Elliotte Rusty Harold. وبكلمة أخرى تشير العبارة إلى أو المورد نحـو http://metalab. unc.edu/xml/ له خاصية Creator وقيمته سلسلة الحرفية هـــي Elliotte Rusty Harold.

مسافات الاسم

يتم استخدام مسافات الاسم للتمييز بين عناصر RDF والعناصر التي توجـــد فـــي أنـــواع وقيـــم الخواص. وعادة ما يستخدم حيز الاســـم http://www.w3.org/1999/02/22-rdf-syntax-ns# لعناصر RDF والتي تكون بادئتها rdf على وجه العموم. وفي المثال السابق يوجد عنصــر Creator في حيز الاسم الافتراضي. وعلى أية حال يمكن "بل ويجب" أن يأتي الوصف من حيز الاسم المختلفة وغير الافتراضي وذلك مثل عنصر RDF في التعليمات البرمجية ١٩-٢ والـــذي يستخدم مصطلحات Dublin Core وحيز الاسم http://purl.org/DC/.

تعليمات البرمجة ١٩-٢: عناصر مصطلحات Dublin Core داخل حيز الاسم

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/DC/">
 <rdf:Description about="http://metalab.unc.edu/xml/">
  <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
 </rdf:Description>
</rdf:RDF>
```

The Dublin Core

يعد "http://purl.org/dc/" الذي يخص Dublin Core مجموعة من العناصر المصممة بمساعدة الباحثين في إيجاد الموارد الإلكترونية بطريقة تشبه كتالوج بطاقات المكتبات. وتضم عناصر Dublin Core معلومات أساسية مفهرسة وخاصة.

- العنوان: وهو الاسم المورد.
- المنشئ: وهو الشخص أو المنظمة التي أنشأته معظم المورد "اسم المؤلف لرواية أو مصور التقط صورة".
- الوصف: وصف مختصر للمورد مثل فكرة.
- الناشر: اسم الشخص أو المنظمة التي تتيح المورد "مثل IDG Books, Claremont University أو Apple Computer".
- المساهم: وهو غير المنشئ ولكن من يساهم في المورد "كالموضح أو محرر الرواية".
- التاريخ: تاريخ جعل المورد متاحاً في شكله الحالي وتأخذ عادة التنسيق YYYY-MM-DD مثل 1999-12-31.
- النوع: وهو نوعية المورد مثل صفحة ويب أو قصة قصيرة أو شعر أو مقالة أو صورة. وما زال العمل جاري لإخراج قائمة محددة لأنواع الموارد المقبولة.
- التنسيق: وهو تنسيق المورد مثل PDF أو HTML أو JPEG ومازال العمل جاري لإخراج قائمة محددة لتنسيقات الموارد المقبولة.
- المعرف: وهو سلسلة أو رقم للمورد مثل "URL أو رقم أمن قوس أو ISBN".
- المصدر: وهو سلسلة أو رقم يعرف وحده العمل الذي نقل عنه المورد. وذلك مثل صفحة الويب لنص من رواية Jerome K. Jerome's في القرن ١٩ Three Men

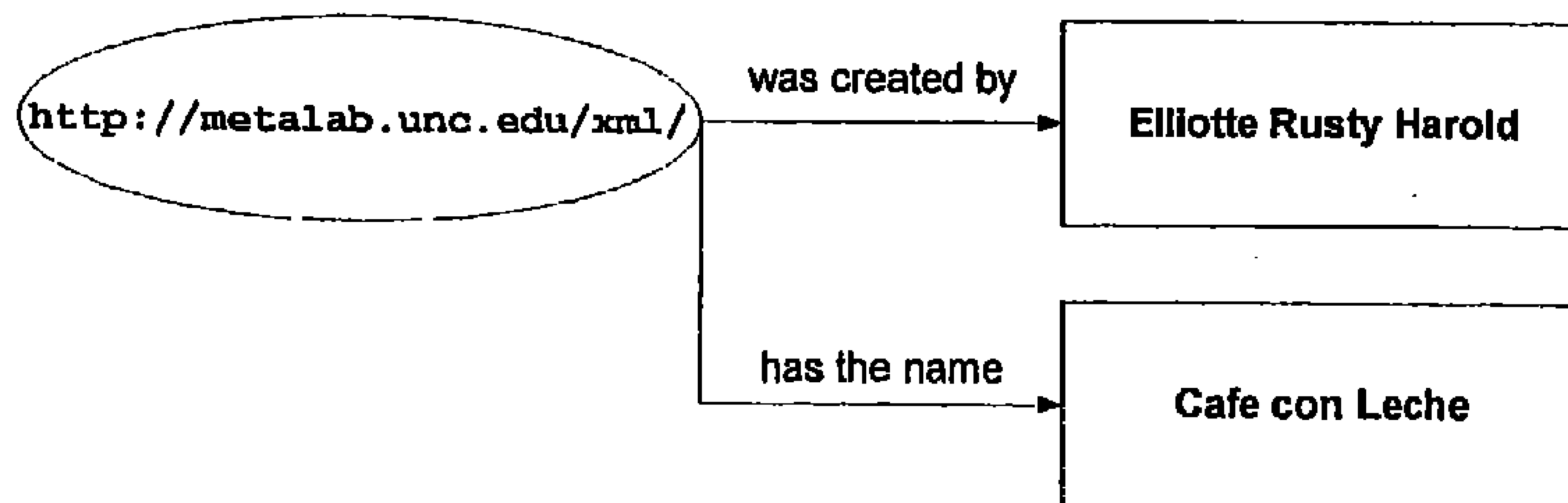
in a Boat والتي قد نستخدم هذا في تحديد الإصدار المعين الذي جـاء منـه هـذا النص.

- ♦ اللغة: أي اللغة الرئيسية التي تكتب بها المورد مثل لغة ISO 639
- ♦ الحقوق وهي حقوق الطباعة وأي ملحوظات تحدد تحت أي ظروف يمكـن اسـتخدام المورد من عدمه.

وتوجد مجموعة أخرى لعناصر Dublin Core المتاحة في مرحلة التجربة متضمنـة العلاقة RELATION والتغطية COVERAGE. ونستخدم Dublin Core في أمثلة هـذا الفصل. وعلى أية حال لا يوجد حد لاستخدام هذه العناصر. فيمكن استخدام مصطلحات ومسافات الاسم المختلفة للخواص طالما تم وضعها في مسافات الاسم.

التعليمات البرمجية والخواص المتعددة

يمكن لعنصر واحد من Description وأن يحدد أكثر من خاصية عن المورد. وذلك مثل ما لـم يذكر في العبارة السابقة وهو اسم الموقع Cafe con Leche. والعبارة التي تتضمن هذا تكتـب هكذا "Elliotte Rusty Harold is the author of the Cafe con Leche Web site at the URL http://metalab.unc.edu/ xml/." أما كتابتها بصيغة RDF فتكون "The Web site at the URL http://metalab.unc.edu/xml/ has the name Cafe con Leche and was created by Elliotte Rusty Harold." والشكل "١٩-٣" يوضح هذه العبارة. أما التعليمات البرمجية ١٩-٣ بتوضيح كيفية إضافة اسم الخاصية إلى سلسلة RDF بطريقة طبيعيـة كتابع أخر لدى rdf:Description, dc:TITLE.



الشكل ١٩-٣ رسم توضيحي لعبارة من التعليمـات البرمجيـة ذات الخـواص المتعددة.

تعليمات البرمجة ١٩-٣: عبارة التعليمات البرمجية ذات الخواص المتعددة في شكل RDF التسلسلي

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/DC/>

  <rdf:Description about="http://metalab.unc.edu/xml/>
    <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
    <dc:TITLE>Cafe con Leche</dc:TITLE>
  </rdf:Description>

</rdf:RDF>
```

ويمكن لعنصر واحد من RDF احتواءه على أي رقم وعناصر Description والذي يسمح بإنشائه أي عدد من العبارات. ذلك كفصل العبارتين "Elliotte Rusty Harold is the author of the Cafe con Leche Web site at the URL http://metalab.unc.edu/xml/" وعبارة "Elliotte Rusty Harold is the author of the Cafe au Lait Web site at the URL http://metalab.unc.edu/ javafaq/." هما هكذا عبارتان عن موردين مختلفين. وتوضيح التعليمات البرمجية ١٩-٤ كيفية وضع التعليمات البرمجية لها في RDF.

تعليمات البرمجة ١٩-٤: عبارتا التعليمات البرمجية RDF

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/DC/">

  <rdf:Description about="http://metalab.unc.edu/xml/">
   <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
   <dc:TITLE>Cafe con Leche</dc:TITLE>
  </rdf:Description>
```

```
<rdf:Description about="http://metalab.unc.edu/javafaq/">
  <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
  <dc:TITLE>Cafe au Lait</dc:TITLE>
</rdf:Description>

</rdf:RDF>
```

الخواص المقيمة للمورد

هناك عبارات أكثر تركيباً مثل "The Cafe con Leche Web site at the URL http://metalab.unc.edu/xml/ has the creator Elliotte Rusty Harold, whose email address is elharo@metalab.unc.edu." والمفتاح هنا هو ما يكون البريد الإلكتروني فهو يشير إلى معين فريد لشخص واحد وخاصة mailto:elharo@ metalab.unc.edu لدى URL. وهكذا يصبح الشخص مورداً عن كونه مجرد حرفي. وهذا المورد هو قيمة لخاصية "created by" نحو المورد http:// metalab.unc.edu/xml/ ويوضح "الشكل ١٩-٤" هذه العبارة.

http://metalab.unc.edu/xml/

was created by

mailto:elharo@metalab.unc.edu

has the name

has the name

Cafe con Leche

Elliotte Rusty Harold

الشكل ١٩-٤ شكل توضيحي لعبارة من التعليمات البرمجية ذات الخواص المقيمة للمورد

ووضع التعليمات البرمجية لهذه العبارة في RDF تكون مباشرة. فببساطة يوضع لعنصر Creator التابع Description والذي يصف عنصر mailto:elharo@metalab كما يظهر في التعليمات البرمجية ١٩-٥.

تعليمات البرمجة ١٩-٥: عبارة من التعليمات البرمجية في RDF مع عناصر Description المبنية

```
<RDF xmlns=http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://www.purl.org/DC/">

  <Description about="http://metalab.unc.edu/xml/">
   <dc:TITLE>Cafe con Leche</dc:TITLE>
   <dc:CREATOR/>
     <Description about="mailto:elharo@metalab.unc.edu">
       <dc:TITLE>Elliotte Rusty Harold</dc:TITLE>
     </Description>
   </dc:CREATOR>
  </Description>
</RDF>
```

ليس هناك أية حدود وذلك لبناء الوصف أو أية حدود لعدد الخواص التي يمكن تطبيقها على عنصر Description سواء كان مبنياً أو غير مبني.

تعطي RDF صيغة أخرى لا يبنى بداخلها عناصر Description وإنما يحتوي المورد الموصوف سمة resource والذي يشير إلى URI لعنصر Description. ومثال ذلك تعليمات البرمجة ١٩-٦ وهي سلسلة معادلة لجملة "The Cafe con Leche Web site at the URL http://metalab.unc.edu/xml/ has the creator Elliotte Rusty Harold, whose email address is elharo@metalab.unc.edu."

تعليمات البرمجة ١٩-٦: استخدام Descriptions سمة المورد

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://www.purl.org/DC/>

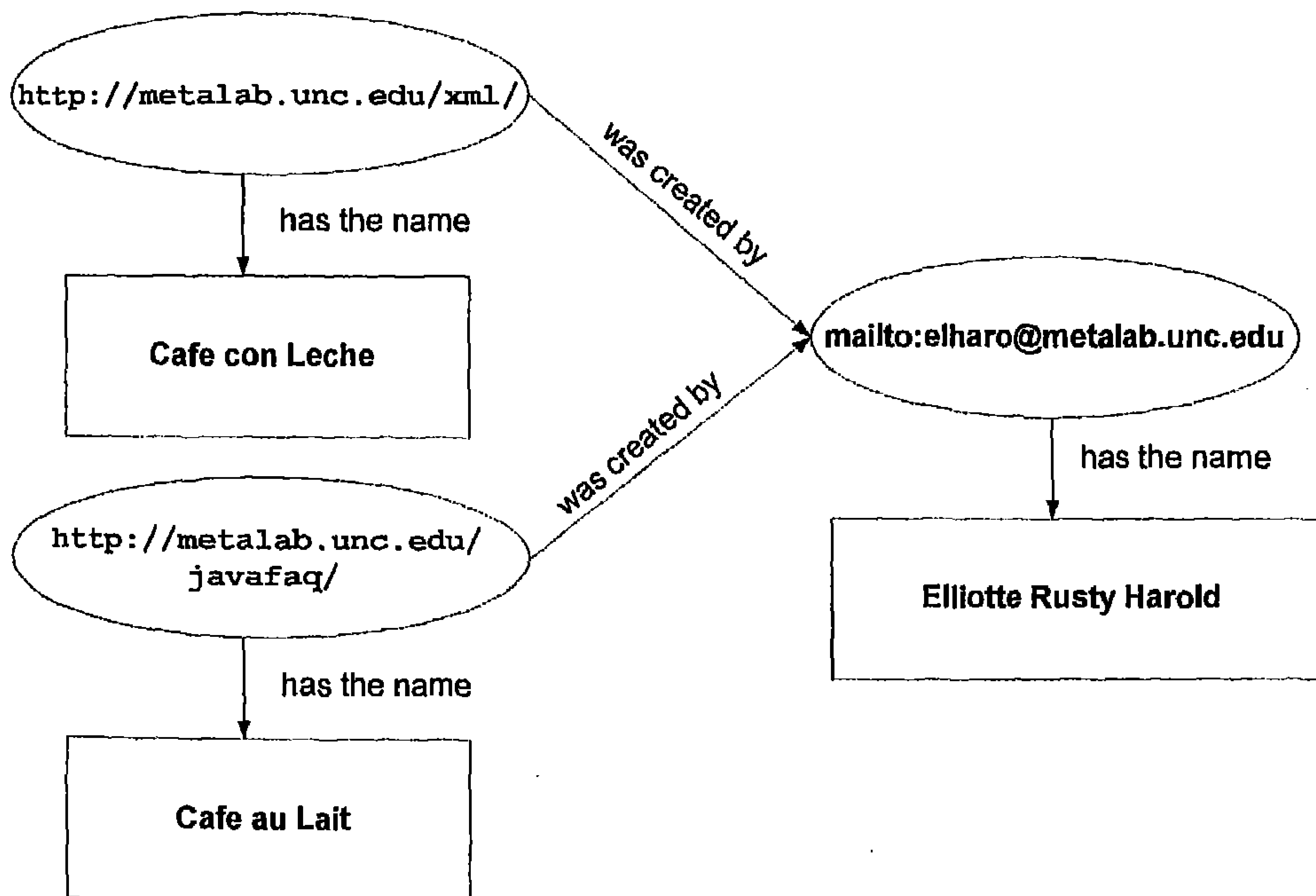
 <rdf:Description about="http://metalab.unc.edu/xml/">
  <dc:TITLE>Cafe con Leche</dc:TITLE>
```

```
 <dc:CREATOR rdf:resource="mailto:elharo@metalab.unc.edu"/>
</rdf:Description>

<rdf:Description about=""mailto:elharo@metalab.unc.edu">
 <dc:TITLE>Elliotte Rusty Harold</dc:TITLE>
</rdf:Description>

</rdf:RDF>
```

وعلى الرغم من أن هذه الصيغة تبدو صعبة الفهم بالنسبة للإنسان، فهي لا تشكل أي صعوبة واردة لبرنامج الكمبيوتر. والميزة الأساسية هي إتاحة نفس الخاصية في إلحاقها بالموارد المتعددة. ومثال ذلك عبارة "Elliotte Rusty Harold, whose email address is elharo@ metalab.unc.edu, created both the Cafe con Leche Web site at the URL http:// metalab.unc.edu/xml/ and the Cafe au Lait Web site at the URL http://metalab. unc.edu/javafaq/".



الشكل ١٩–٥ شكل توضيحي لعبارة التعليمات البرمجية.

تعليمات البرمجة ١٩-٧: عبارة التعليمات البرمجية مع نفس الخاصية الملحقة بالموارد المتعددة

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://www.purl.org/DC/">

 <rdf:Description about="http://metalab.unc.edu/xml/">
  <dc:TITLE>Cafe con Leche</dc:TITLE>
  <dc:CREATOR rdf:resource="mailto:elharo@metalab.unc.edu"/>
 </rdf:Description>

 <rdf:Description about="http://metalab.unc.edu/javafaq/">
  <dc:TITLE>Cafe au Lait</dc:TITLE>
  <dc:CREATOR rdf:resource="mailto:elharo@metalab.unc.edu"/>
 </rdf:Description>

 <rdf:Description about="mailto:elharo@metalab.unc.edu">
  <dc:TITLE>Elliotte Rusty Harold</dc:TITLE>
 </rdf:Description>

</rdf:RDF>
```

خواص لغة XML المقيمة

تنتشر قيم الخواص سواء كان نصاً نقياً أو مورداً وعلى أية حال. قد تحتوي القيم على تريد جيد التنسيق للغة XML وهو ليس نفسه ترميز RDF. وفي هذه الحالة يجب على عنصر الخاصية أن يكون لديه سمة parseType مع القيمة Literal كما تظهر في التعليمات البرمجية ١٩-٨.

تعليمات البرمجة ٨-٨: SEASON يحتوي على تابعي LEAGUE

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
 xmlns:dc="http://www.purl.org/DC/
 xmlns:nm="http://www.metalab.unc.edu/xml/names/">
```

```
<rdf:Description about="http://metalab.unc.edu/xml/">
  <dc:CREATOR parseType="Literal">
    <nm:FirstName>Elliotte</nm:FirstName>
    <nm:MiddleName>Rusty</nm:MiddleName>
    <nm:LastName>Harold</nm:LastName>
  </dc:CREATOR>
</rdf:Description>

</rdf:RDF>
```

وبدون =parseType"literal" يجب أن تكون قيمة الخاصية فقط مورداً أو بيانات رمزية مفهومة. فلا يجب أن تحتوي على أي ترميز متضمن لها.

صيغة RDF المختصرة

ويمثل استخدام الصيغة الأساسية عالية يقوم RDF بتعريف صيغ مختصرة تستخدم السمات بدلاً من محتوى البيانات الرمزية المفهومة. وهذا يتلاءم أكثر عندما تضمن بيانات RDF مع صفحة HTML لأن متصفح الويب يمكن ببساطة أن يتجاهل علامات RDF دون أية تأثير على الصفحة الموجودة. وكل من الصيغتين تتساوى تماماً من جانب موزع RDF "بعكس HTML".

وتصبح كل خاصية. في الصيغة المختصرة. سمة لعنصر Description واسم الخاصية هو اسم السمة. أما إذا كانت للخاصية قيمة حرفية فتكون قيمة الخاصية هي قيمة السمة. وإذا كانت للخاصية قيمة مورد فإن الخاصية تكون URI وذلك بالنسبة للمورد كما يصف عنصر Description المنفصل المورد. ولأن عنصر Description ليس له أي تنوع للعناصر التابعة فهو لا يحتاج إلى علامة انتهاء ويكتب باستخدام صيغة عنصر عادي وفارغ.

وتكتب عبارة "Elliotte Rusty Harold created the Web site http://metalab.unc.edu/xml/" بطريقة مختصرة كما يلي:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/DC/">
  <Description about="http://metalab.unc.edu/xml/"
    dc:CREATOR="Elliotte Rusty Harold" />
</RDF>
```

كما تكتب عبارة "Elliotte Rusty Harold created the Cafe con Leche Web site http:// metalab.unc.edu/xml/" بطريقة مختصرة كما يلي:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc='http://purl.org/DC/">
  <Description about="http://metalab.unc.edu/xml/
   dc:CREATOR="Elliotte Rusty Harold"
   dc:TITLE="Cafe con Leche" />
</RDF>
```

أما بالنسبة لخاصية المورد المقيمة فقد تخدعنا في اختصارها. فعبـــارة "The Cafe con Leche Web site at the URL http://metalab.unc.edu/xml/ has the creator Elliotte Rusty Harold, whose email address is elharo@metalab.unc.edu" يمكن أن تختصر هكذا:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/DC/">
  <rdf:Description about="http://metalab.unc.edu/xml/"
   dc:TITLE="Cafe con Leche>
    <dc:CREATOR rdf:resource="mailto:elharo@metalab.unc.edu"
     dc:TITLE="Elliotte Rusty Harold" />
  </rdf:Description>
</rdf:RDF>
```

وهنا لا يعد عنصر Description فارغاً لأن له التابع Creator وعلى أية حال لا يحتـــوي هذا العنصر على أية بيانات رمزية ما عدا الفراغ الأبيض.

الحاويات

عندما يصف عنصر RDF مورداً له قيم عديدة من نفس النوع وذلك مثل قيام عدة أشخاص بكتابة مستند ما أو القيام بإدراج المواقع المتشابهة والتي يمكن من عليها إيجاد صفحة الويـــب فيمكـــن للحاوية جمع قيم الخواص. وتعد كل وحدة في المجموعة قيمـــة خاصيـــة لنفـــس الفـــرع "اســـم الخاصية". وهذا يتيح الوصف للمجموعة ككل بدلاً من وصف الوحدات الفردية فـــي الحاويـــة. ويعرف RDF ثلاثة أنواع من كائنات الحاوية:

١- **Bag**: وهو مجموعة غير مرتبة الخواص.

٢- **Seq**: إدراج مرتب للخواص.

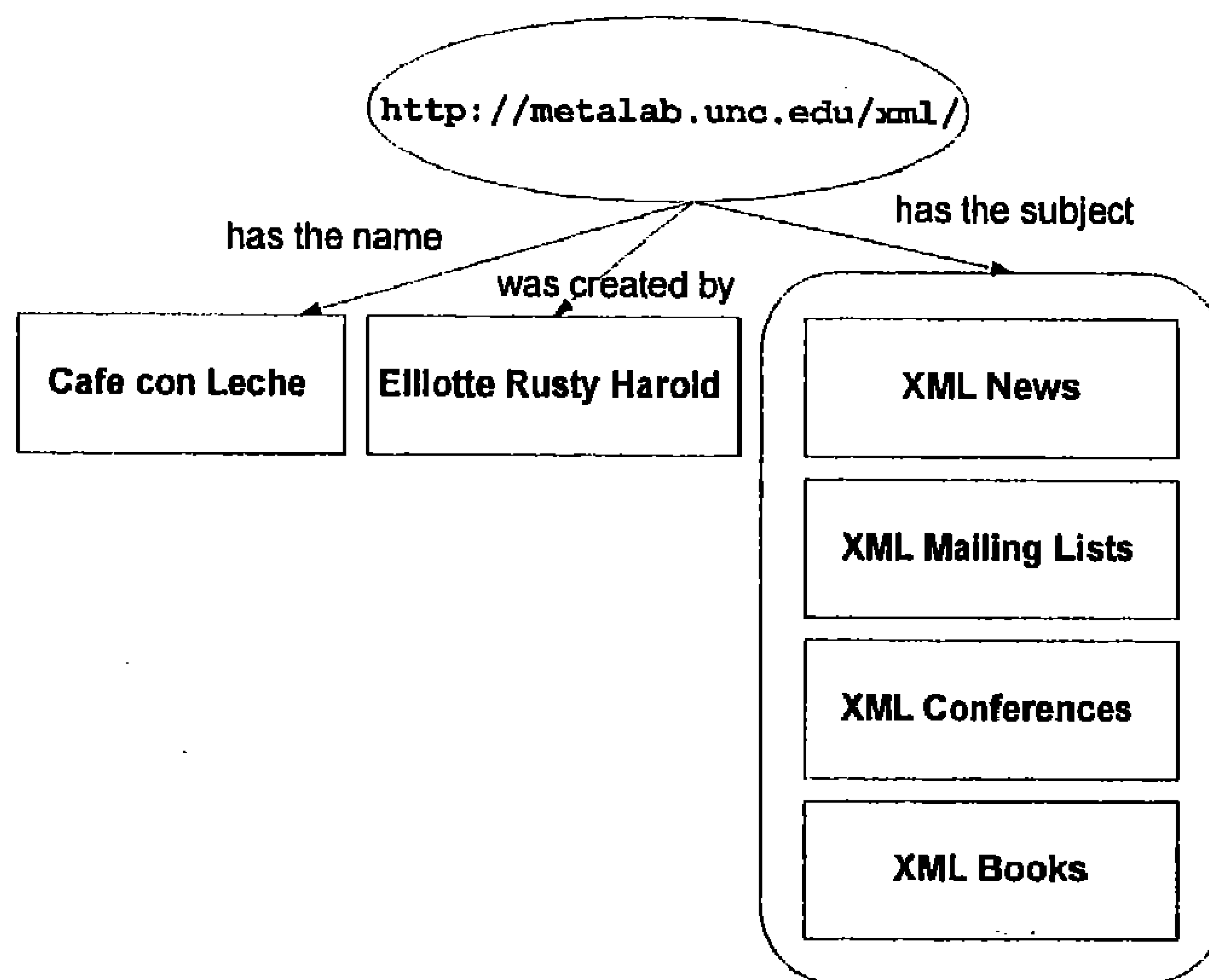
٣- **Alt**: إدراج لخواص بديلة والتي يمكن من خلالها اختيار واحدة.

حاوية Bag

وهي إدراج لقيم الخواص "الموارد والأحرف" دون ترتيب معين وكل منها يساهم في نفس الاسم "النوع" للخاصية. وهذا يتيح إظهار خاصية لها أكثر من قيمة مثل مؤلفي الكتاب أو أعضاء لجنة. وقد يحتوي Bag على قيمتين مزدوجتين.

يتم تقديم مجموعة الخواص عن طريق عنصر Bag. وتعد كل وحدة بها عنصراً تابعاً "li" تابعاً لعنصر "Bag". أما Bag نفسها فهي عنصر تابع لعنصر Description والتي تطبق عليه.

مثال ذلك عبارة "The Cafe con Leche Web site at http://metalab.unc.edu/xml/ was created by Elliotte Rusty Harold to provide XML news, XML mailing lists, XML conferences, and XML books" وهي موضحة في الشكل ١٩-٦". ويمكن جمع الأربعة عناصر الأساسية في Bag كما هو موضح في التعليمات البرمجية ١٩-٩.



الشكل ١٩-٦ شكل توضيحي لعبارة التعليمات البرمجية

تعليمات البرمجة ١٩-٩: أربعة أعضاء في حاوية المجموعة

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
 xmlns:dc="http://www.purl.org/DC#">

 <rdf:Description about="http://metalab.unc.edu/xml/">
   <dc:TITLE>Cafe con Leche</dc:TITLE>
   <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
   <dc:SUBJECT>
     <rdf:Bag>
       <rdf:li>XML News</rdf:li>
       <rdf:li>XML Mailing lists</rdf:li>
       <rdf:li>XML Conferences</rdf:li>
       <rdf:li>XML Books</rdf:li>
     </rdf:Bag>
   </dc:SUBJECT>
 </rdf:Description>

</rdf:RDF>
```

وإذا كانت عناصر حاوية bag موارد عن كونها أحرف فيتم مع سمة resources. ومثـــال ذلك التعليمات البرمجية ١٩-١٠ والتي تعطي خريطة لموقع بسيط وهو Cafe con Leche.

تعليمات البرمجة ١٩-١٠: خريطة الموقع البسيط Cafe con Leche في حاوية Bag

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://www.purl.org/DC#">

 <rdf:Description about="http://metalab.unc.edu/xml/">
   <dc:TITLE>Cafe con Leche</dc:TITLE>
   <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
   <dc:SUBJECT>
     <rdf:Bag>
       <rdf:li
         resource="http://metalab.unc.edu/xml/news1999.html"/>
       <rdf:li
     resource="http://metalab.unc.edu/xml/mailinglists.html/>
       <rdf:li
         resource="http://metalab.unc.edu/xml/news1999.html"/>
```

```
      <rdf:li
      resource="http://metalab.unc.edu/xml/tradeshows.html"/>
    </rdf:Bag>
  </dc:SUBJECT>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/news1999.html">
  <dc:TITLE>XML News from 1999</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/books.html">
  <dc:TITLE>XML Books</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/mailinglists.html">
  <dc:TITLE>XML Mailing Lists</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/tradeshows.html">
  <dc:TITLE>XML Trade Shows and Conferences</dc:TITLE>
</rdf:Description>

</rdf:RDF>
```

حاوية Seq

تشبه حاوية التسلسل حاوية المجموعة. ولكنها تضمن لنا ترتيب المحتويات. وهي تكتب كما تكتب المجموعات إلا أن عنصر Seq يحل محل عنصر Bag. وذلك مثل ضمان هذا التسلسل الخروج بالترتيب الآتي عندما يقرأ موزع RDF تعليمية Subject نحو "XML News, XML Mailing Lists, XML Conferences, XML Books" وليس أي ترتيب آخر مثل: "XML Books, XML Conferences, XML Mailing Lists, XML News".

```
  <dc:SUBJECT>
    <rdf:Seq>
```

```
    <rdf:li>XML News</rdf:li>
    <rdf:li>XML Mailing lists</rdf:li>
    <rdf:li>XML Conferences</rdf:li>
    <rdf:li>XML Books</rdf:li>
  </rdf:Seq>
</dc:SUBJECT>
```

ويتضح بالتطبيق أن ترتيب الخواص في المحتوى ليس بالأهمية الضرورية ولذا لا تستخدم هذه الحاويات مثل حاوية المجموعات وحاوية البدائل.

حاوية Alt

وتحتوي على عضو أو أكثر يتم من خلالهم اختيار واحداً. وذلك مثل استخدامها في وصف البدائل على موقع الويب. وفي العبارة التالية توضح المواقع البديلة في إدراج البدائل لأن واحداً فقط هو المراد. The Cafe au Lait Web site at http://metalab.unc.edu/javafaq/ created by Elliotte Rusty Harold is mirrored at Sunsite Austria (http://sunsite. univie.ac.at/jcca/mirrors/javafaq/), Sunsite Slovakia (http://sunsite. uakom.sk/javafaq/), Sunsite Sweden (http://sunsite.kth.se/javafaq/) و and Sunsite Switzerland (http://sunsite.cnlab-switch.ch/javafaq/)
والتعليمات البرمجية ١٩-١١ توضح تسلسل RDF.

تعليمات البرمجة ١٩-١١: المواقع البديلة لموقع Cafe au Lait في حاوية Seq

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://www.purl.org/DC#">

  <rdf:Description about="http://metalab.unc.edu/xml/">
    <dc:TITLE>Cafe con Leche</dc:TITLE>
    <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
    <dc:PUBLISHER>
      <rdf:Alt>
        <rdf:li resource =
         "http://sunsite.univie.ac.at/jcca/mirrors/javafaq/" />
        <rdf:li resource =
```

```
      "http://sunsite.kth.se/javafaq/" />
     <rdf:li resource =
      "http://sunsite.cnlab-switch.ch/javafaq/" />
     <rdf:li resource =
      "http://sunsite.uakom.sk/javafaq/" />
   </rdf:Alt>
  </dc:PUBLISHER>
 </rdf:Description>

 <rdf:Description
  about="http://sunsite.univie.ac.at/jcca/mirrors/javafaq/">
  <dc:PUBLISHER>Sunsite Austria</dc:PUBLISHER>
 </rdf:Description>

 <rdf:Description
  about="http://sunsite.uakom.sk/javafaq/">
  <dc:PUBLISHER>Sunsite Slovakia</dc:PUBLISHER>
 </rdf:Description>

 <rdf:Description
  about="http://sunsite.cnlab-switch.ch/javafaq/">
  <dc:PUBLISHER>Sunsite Switzerland</dc:PUBLISHER>
 </rdf:Description>

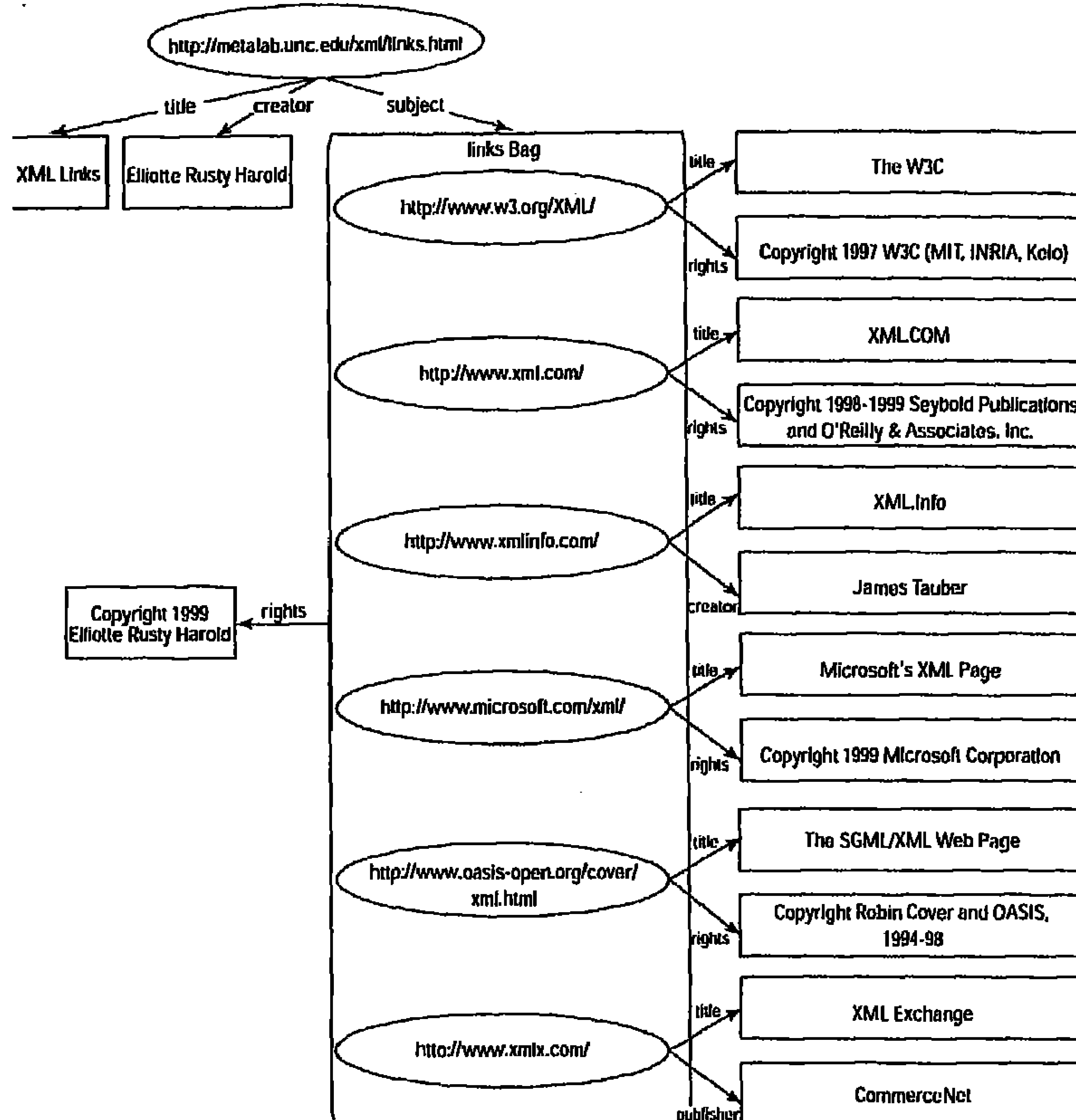
 <rdf:Description
  about="http://sunsite.kth.se/javafaq/">
  <dc:PUBLISHER>Sunsite Sweden</dc:PUBLISHER>
 </rdf:Description>

</rdf:RDF>
```

تعليمات برمجية عن الحاويات

يمكن عمل التعليمات البرمجية عن الحاويات بشكل كلي بعيداً عن التعليمات الخاصة بـالوحدات المنفردة للحاويات. فقد ترغب في الإشارة إلى شخص معين قام بتطوير موقع الويب دون التلميح إلى أنه قد قام بكتابة كل صفحة على الموقع. أو أنك تطلب حقوق الطبع لمجموعة ارتباطات دون المطالبة بحقوق الطبع للصفحات التي تقوم أنت بربطها. "مثال ذلك قيم السوق لمجموعة ارتباطات ووصف Yahoo وذلك لفئات الملايين من الدولارات بالرغم من عدم امتلاك Yahoo

لأي من الصفحات المرتبطة بها" وفي الواقع قد يكون لأعضاء الحاوية حقوق طبع مختلفة عن الحاوية نفسها. ويوضح شكل ١٩-٧ هذا.



الشكل ١٩-٧ اختلاف معلومات حقوق الطبع لحاوية المجموعة عن تلك لأعضاء حاوية المجموعة وحدها.

ولوضع التعليمات البرمجية هذه في RDF يتم إعطاء حاوية "Bag أو Seq أو Alt" سمة ID. وتقوم عناصر Description ذات سمات about والتي ترتبط قيمها مع URL التي تشير إلى الحاوية بوصف الحاوية.

تعليمات البرمجة ١٩-١٢: وضع التعليمات البرمجية لوصف الحاوية RDF

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://www.purl.org/DC#">
```

```
<rdf:Description
 about="http://metalab.unc.edu/xml/links.html">
 <dc:TITLE>XML Links</dc:TITLE>
 <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
 <dc:SUBJECT>
  <rdf:Bag ID="links">
   <rdf:li resource="http://www.w3.org/XML/"/>
   <rdf:li resource="http://www.xml.com/"/>
   <rdf:li resource="http://www.xmlinfo.com/"/>
   <rdf:li resource="http://www.microsoft.com/xml//>
   <rdf:li
    resource="http://www.oasis-open.org/cover/xml.html"/>
   <rdf:li resource=http://www.xmlx.com//>
  </rdf:Bag>
 </dc:SUBJECT>
</rdf:Description>

<rdf:Description about="#links">
 <dc:RIGHTS>
  Copyright 1999 Elliotte Rusty Harold
 </dc:RIGHTS>
</rdf:Description>

<rdf:Description about="http://www.w3.org/XML/">
 <dc:TITLE>The W3C</dc:TITLE>
 <dc:RIGHTS>
  Copyright 1997 W3C (MIT, INRIA, Keio)
 </dc:RIGHTS>
</rdf:Description>

<rdf:Description about="http://www.xml.com/">
 <dc:TITLE>xml.com</dc:TITLE>
 <dc:RIGHTS>
  Copyright 1998-1999 Seybold Publications
  and O'Reilly &amp; Associates, Inc.
 </dc:RIGHTS>
</rdf:Description>
```

```
<rdf:Description about="http://www.xmlinfo.com/">
  <dc:TITLE>XML Info</dc:TITLE>
  <dc:CREATOR>James Tauber</dc:CREATOR>
</rdf:Description>

<rdf:Description about="http://www.microsoft.com/xml/">
  <dc:TITLE>Microsoft's XML Page</dc:TITLE>
  <dc:RIGHTS>Copyright 1999 Microsoft Corporation</dc:RIGHTS>
</rdf:Description>

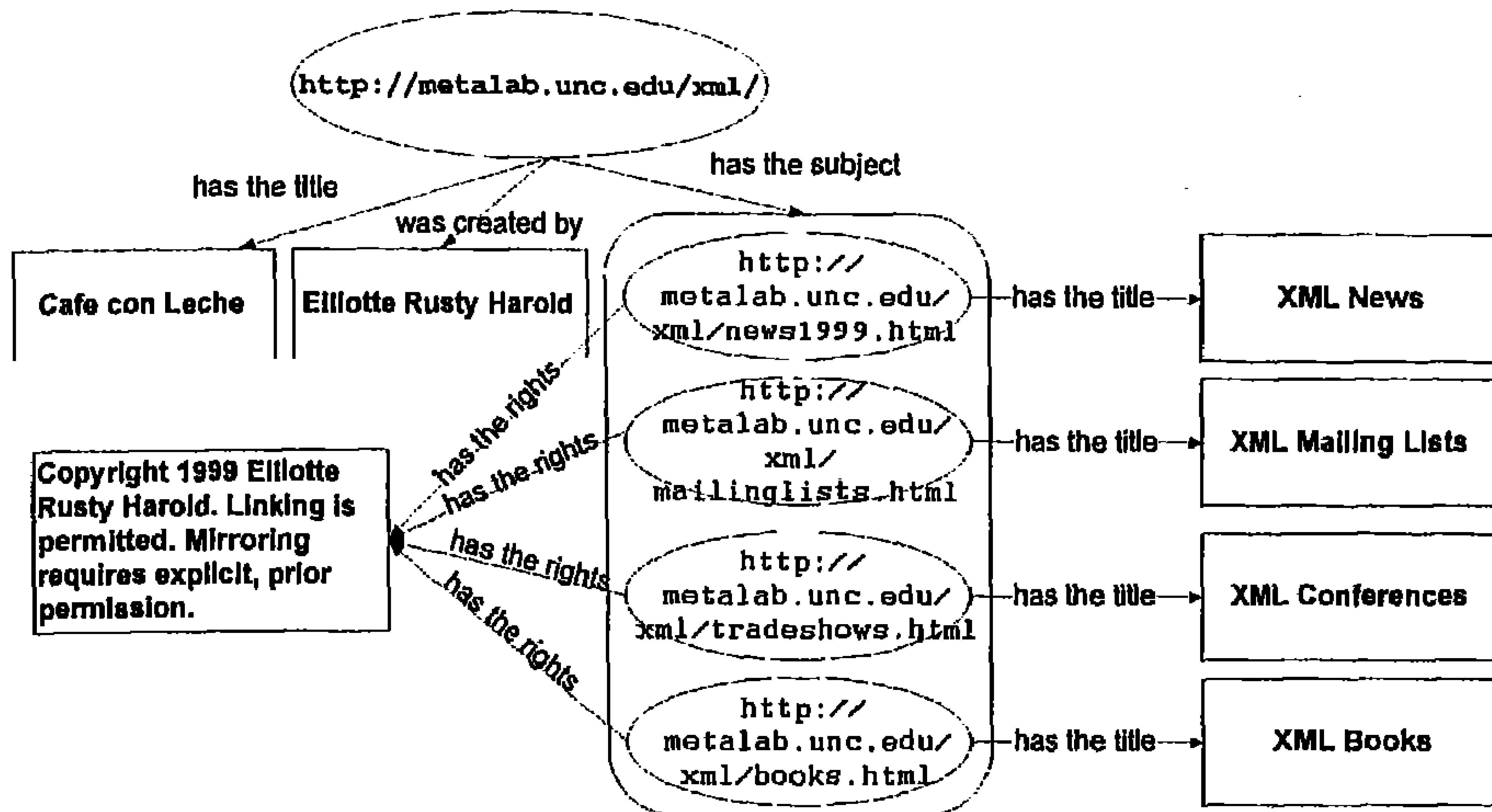
<rdf:Description
  about="http://www.oasis-open.org/cover/xml.html">
  <dc:TITLE>Robin Cover's XML Web Page</dc:TITLE>
  <dc:RIGHTS>
    Copyright Robin Cover and OASIS, 1994-98
</dc:RIGHTS>
</rdf:Description>

<rdf:Description about="http://www.xmlx.com/">
  <dc:TITLE>XML Exchange</dc:TITLE>
  <dc:PUBLISHER>CommerceNet</dc:PUBLISHER>
</rdf:Description>

</rdf:RDF>
```

التعليمات البرمجية لأعضاء الحاوية

في بعض الأحيان يريد المستخدم عمل تعليمات برمجية لكل عضو من أعضاء الحاوية ولكن دون تكرار نفس الوصف ثلاث أو أربع مرات ومثال ذلك تحديد العنوان والمنشئ لكل من المواقع البديلة على أنه Cafe au Lait وElliotte Rusty Harold على التوالي كما هو واضح في "الشكل ١٩–٨".



الشكل ١٩-٨ إرفاق نفس الوصف إلى كل صفحة

ويمكن ضم سمة aboutEach في عنصر Bag أو Seq أو Alt والتي تعـد قيمتـها اسـماً ويمكن من خلاله تطبيق الوصف على جميع أعضاء الحاوية. وذلك مثل تطبيق ملحوظـة عـن حقوق الطبع لكل صفحة من Bag وتوضح التعليمات البرمجية ١٩-١٣ هذا.

تعليمات البرمجة ١٩-١٣: وصف كل عنصر في حاوية Bag

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
 xmlns:dc="http://www.purl.org/DC#">

  <rdf:Description about=""http://metalab.unc.edu/xml/">
   <dc:TITLE>Cafe con Leche</dc:TITLE>
   <dc:CREATOR>Elliotte Rusty Harold</dc:CREATOR>
   <dc:SUBJECT>
    <rdf:Bag aboutEach="pages">
     <rdf:li
      resource="http://metalab.unc.edu/xml/news1999.html"/>
     <rdf:li
    resource="http://metalab.unc.edu/xml/mailinglists.html"/>
     <rdf:li
      resource="http://metalab.unc.edu/xml/news1999.html"/>
```

```
    <rdf:li
    resource="http://metalab.unc.edu/xml/tradeshows.html"/>
   </rdf:Bag>
  </dc:SUBJECT>
</rdf:Description>

<rdf:Description aboutEach="#pages">
  <dc:RIGHTS>
   Copyright 1999 Elliotte Rusty Harold
   Linking is permitted.
   Mirroring requires explicit, prior permission.
  </dc:RIGHTS>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/news1999.html>
  <dc:TITLE>XML News from 1999</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/books.html">
  <dc:TITLE>XML Books</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/mailinglists.html">
  <dc:TITLE>XML Mailing Lists</dc:TITLE>
</rdf:Description>

<rdf:Description
  about="http://metalab.unc.edu/xml/tradeshows.html">
  <dc:TITLE>XML Trade Shows and Conferences</dc:TITLE>
</rdf:Description>

</rdf:RDF>
```

التعليمات البرمجية للمجموعة المتضمنة

يمكن للمستخدم عمل تعليمات برمجية على مجموعة للموارد قد تكون مجموعة من نفس الحاوية أو لا. فمثلاً عند الحاجة إلى التحديد بأن كل صفحة على موقع الويب http://www.macfaq.com is"Copyright 1999 Elliotte Rusty Harold" فيمكنك عمل هذا عن طريق عنصر Description والذي ينطبق على كل الموارد التي يبدأ URI الخاص بها بسلسلة http://www.macfaq. com ويجب على Description أن يكون لديه سمة aboutEachPrefix والتي تكون قيمتها هي البادئة URI بالنسبة للموارد التي ينطبق عليها الوصف وذلك مثل:

```
<rdf:Description aboutEachPrefix="#http://www.macfaq.com">
  <dc:RIGHTS>Copyright 1999 Elliotte Rusty Harold</dc:RIGHTS>
</rdf:Description>
```

وينشئ عنصر Description مجموعة ضمنية يعد أعضاءها هم الموارد التي تطابق البادئة. وقد تكون هذه الموارد أعضاء حاويات أخرى في ملف RDF وقد تكون عناصر متقاربة وتجمع أعضاء هذه المجموعة المتضمنة من أي مكان توجد به.

وتستخدم بادئات URI فقط في اختيار شجرة ثانوية لموقع الويب. وذلك مثل طلب الوصف لكل الصفحات في metalab.unc.edu على شكل xml/ الهيكلي بأن تكون "Copyright 1999 Elliotte Rusty Harold" وعلى أية حال فهي لا تنطبق على الصفحات الأخرى خارج الشكل الهيكلي نحو:

http://metalab.unc.edu/id/asiasylum أو http://metalab.unc.edu/stats/.

```
<rdf:Description
  aboutEachPrefix="#http://metalab.unc.edu/xml/">
  <dc:RIGHTS>Copyright 1999 Elliotte Rusty Harold</dc:RIGHTS>
</rdf:Description>
```

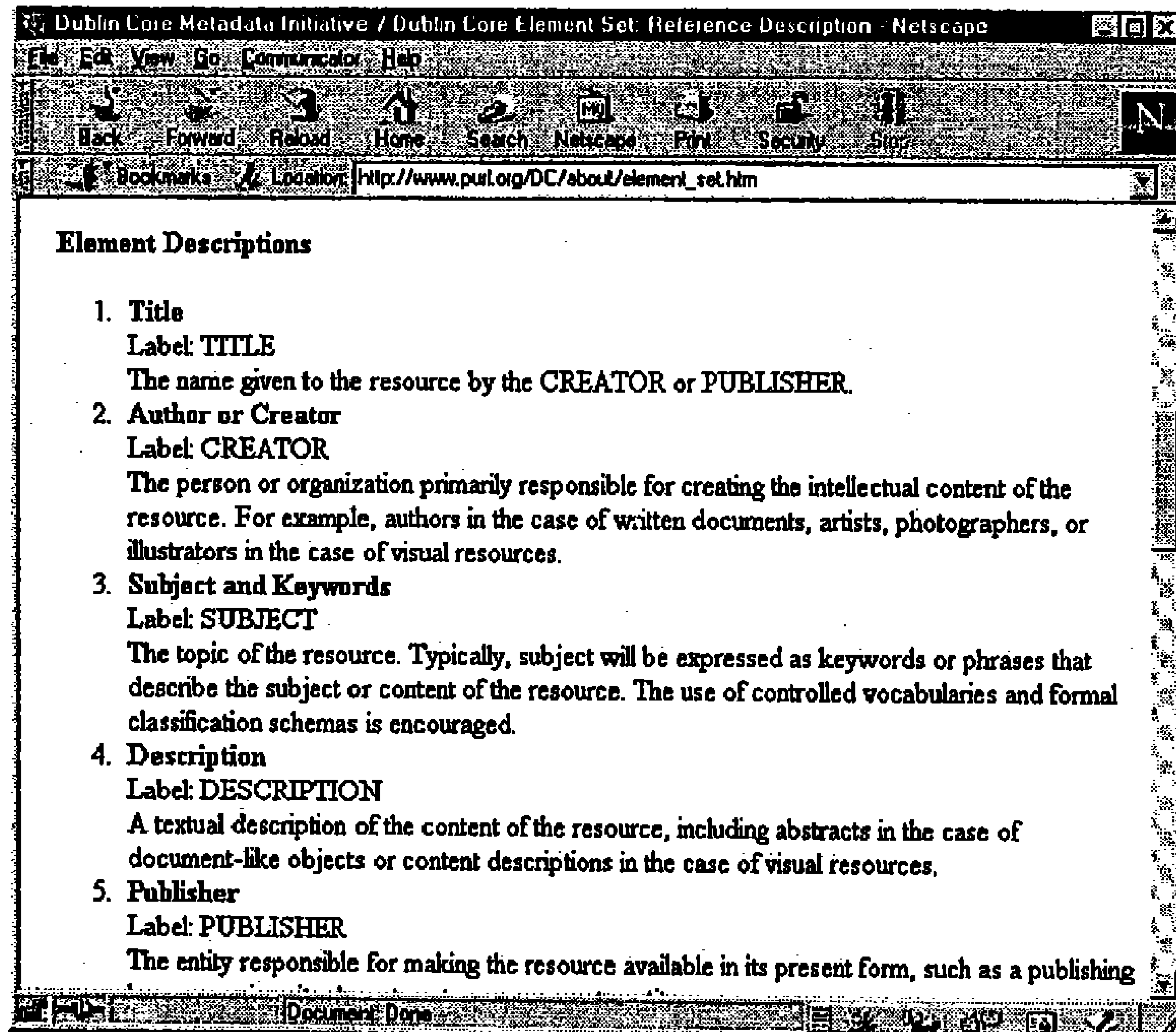
ومثال أخر ISBNs التي يحددها الناشرون. فكل الكتب التي تنشرها IDG Books لها ISBN يبدأ بالرقم 07645. وهكذا قوم عنصر Description بإنشاء Bag يحتوي على الكتب التي تنشر عبر IDG Books كما يقوم بتحديد خاصية property لكل عضو.

```
<rdf:Description aboutEachPrefix="#urn:isbn:07645">
  <dc:PUBLISHER>IDG Books</dc:PUBLISHER>
</rdf:Description>
```

مخططات RDF

وعلى الرغم من عدم ضمان إشارة حيز الاسم العام URI للغة XML إلى أي شيء علـــى وجـــه الخصوص إلا أن RDF أن يشير إلى مخطط قواعد البيانات بالنسبة لمصطلح. ويصف مخطـــط قواعد البيانات المعاني والصيغ المتاحة لعنصر معين. وذلك مثل إشارة المخطط إلـــى ضـــرورة وجود عنصر DATE في الشكل 31-12-1999 وليس في الشكل December 31, 1999. وقـــد يعمل المخطط قواعد البيانات تعليمات برمجية على شكل DTD وذلك مثل ضرورة احتواء كـــل عنصر BOOK لواحد أو أكثر من عناصر AUTHOR التابعة.

وتعد الكيفية في تكوين مخطط قواعد البيانات لتعليمات برمجيـــة كـــهذا موضوعـــاً للجـــدل فبالتطبيق يتضح أن مخططات RDF الحالية تكتب عادة بشكل نثري يمكن للمســـتخدم قراءتـــه. ومثال ذلك جزء من "schema" Dublin Core الموضحة بشكل ١٩-٩ أو على المدى البعيـــد هناك احتمال لتطوير مخطط قواعد البيانات الذي يخص Dublin Core بشكل رســـمي وكـــامل أكثر.



الشكل ١٩-٩ مخطط قاعدة بيانات Dublin Core.

ويكتب مخطط قواعد البيانات النهائي بصيغة رسمية أكثر حتى تستطيع أجهزة الكمبيوتر استيعابها. وعلى وجه الخصوص تحاول W3C RDF Schema Working Group تطوير بكتابة مخطط قواعد البيانات في RDF في التحقق من صحة مستند معين في RDF مقابل مخططات قواعد البيانات التي تستخدمها. وعلى أية حال سيأخذ هذا العمل بعض الوقت. فإذا كنت في حاجة إليه يمكنك الإطلاع على مخطط RDF لقواعد البيانات http://www.w3.org/TR/1998/WD-rdf-schema/.

خلاصة

يغطي هذا الفصل RDF وعلى الأخص:

- نظام وصف المورد "RDF" على إنه تطبيق بلغة XML للبيانات التفصيلية معلومات عن معلومات.
- يعمل مستند أو عنصر RDF التعليمات البرمجية عن الموارد.
- كل عبارة من التعليمات البرمجية تقوم بتعيين مورداً وخاصية هذا المورد وقيمته.
- يتكون المورد من أي شيء له (URI) Uniform Resource Identifier وتعد مجموعة URLالصفحات الويب شكل واحد في URI.
- قد تكون قيمة الخاصية نصاً عادياً أو مورداً أخراً أو ترميز للغة XML.
- العنصر الجذري لمستند RDF هو RDF.
- يحتوي عنصر RDF على عناصر Description تقوم بعمل التعليمات البرمجية عن الموارد.
- يحتوي كل عنصر Description إما على خاصية حرفية أو سمة resource والتي تكون قيمتها URI لقيمة الخاصية.
- يعرف RDF أيضا الصيغة المختصرة والتي تستبدل فيها الخواص بسمات لنفس الاسم على عنصر Description.
- تمد عناصر Bag أو Seq أو Alt بحاويات للموارد المتعددة ويمكن تطبيق الخواص على الحاوية بشكل كلي أو على العناصر الفردية للحاوية أو كليهما معا.
- يجب على حيز الاسم URI بالنسبة لكل مصطلح تستخدم في مستند RDF أن يشير إلى مخطط قواعد البيانات بالنسبة للمصطلح.

يبدأ الفصل القادم في شرح عدد آخر من تطبيقات لغة XML. وهو يبدأ بتحليل عميق لبرنامج Voyager HTML-in-XML DTD وذلك لتنمية قدراتك في قراءة مجموعة DTDs والتي قام بكتابتها آخرون.

الجزء

٥

في هذا الجزء

الفصل ٢٠

[illegible]

الفصل ٢١

[illegible]
CDF

الفصل ٢٢

[illegible]

الفصل ٢٣

[illegible] XML [illegible]

تطبيقات XML

٢٠

قراءة تعريفات نوع المستند

دائماً ما تكون اللغات التي تم إنشاؤها باستخدام لغة LMX مليئة بالأمثلة التي توضح للمستخدم المعنى الأصلي واستخدام كل عنصر وكل سمة من السمات. ومثل أغلب المبرمجين، فإن أغلب كتاب DTD، ويرمز بها إلى تعريفات نوع المستند، يعتبرون المستندات شيء غير ضروري، ومن الأفضل أن تترك للكاتب الفني إذا كانت هناك ضرورة لها أصلاً. ولذلك فإنه ليس غريباً أن تكون DTD التي تحتوي على المستندات الكافية هي الاستثناء وليس القاعدة، ومن ثم فإنه من الضروري أن تتعلم كيفية قراءة DTD التي كتبها كتاب آخرون.

وهناك أحد الأسباب الأخرى المهمة لتعلم قراءة DTD حيث أنك قد تتعلم الكثير من الحيل والتقنيات التي تستطيع أن تطبقها فيDTD الخاص بك، مثل استخدام كينونات المعاملات الخاصة بقوائم السمات العامة والتي تمكنك قراءة DTD الخاص بالآخرين من تعلمها على نحو جيد.

ويتم فصل DTD التي تخص XHTML من W3C في هذا الفصل. وهي في غاية التعقيد غير أنها مكتوبة على نحو جيد. وعن طريق دراستها بتعمق يمكنك أن تتعلم الكثير من التقنيات المفيدة التي تستطيع أن تنفيذها في DTD الخاص بك. وسوف نوضح المزايا والعيوب التي قد قام بها هؤلاء المصممون. كما سوف نوضح كيفية عمل نفس الشيء بعدة طرق والمزايا والعيوب التي تنتج عن ذلك. وبالإضافة إلى ذلك فإننا سوف نعرض للحيل الشائعة في XML DTD وبعض التقنيات التي يمكن أن تستخدمها في إنشاء DTD الخاص بك.

أهمية قراءة تعريف نوع المستند

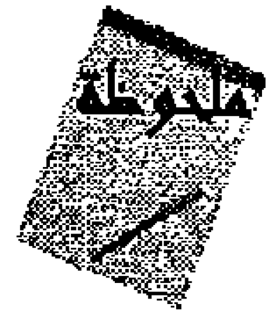
يتم تعريف بعض تطبيقات XML عن طريق مستندات قياسية. ويعد MathML من أحد هذه التطبيقات. حيث استغرق إخراج مثل هذا التطبيق العديد من السنوات والمجهودات المضنية التي قامت على مراحل متعددة من المراجعة الدقيقة، كما كان القائمون على هذا العمل متحفزون لمحاولة التصدي لأي نوع من المشاكل سواء في اللغة أو الوثائق الخاص بهذه اللغة.مما أدى إلى توفير DTD كامل بالإضافة إلى العديد من المعلومات الهامة.

كما أنه لم يتم توثيق كل تطبيقات XML، حيث أن مايكروسوفت قد قامت بإنشاء CDF الذي يتم عرضه في الفصل ٢١. وقد تم توثيق CDF بشكل غير رسمي على Microsoft's Site Builder Network في مجموعة من صفحات الويب التي تم تنظيمها على نحو رديء، غير أنه لا يوجد DTD متاح حالياً. وربما تقوم مايكروسوفت بتحديث وربما بالإضافة إلى CDF، غير أن هذه التحديثات ما زالت غير معروفة حتى الآن.

ومن أحد اللغات التي نادراً ما نجد لها توثيقاً هي لغة الترميز الكيميائية Chemical Marckup Language والتي يرمز لها بالرمز CML وقد ابتكرها بيتر موراي راست. وهي تحتوي على DTD غير أنها تترك العديد من النقاط للمستخدم. فعلى سبيل المثال

تحتوي هذه اللغة على عنصر bondArray غير أن المعلومة الوحيدة عن هذا العنصر هي أنـــه يحتوي على CDATA، ولا توجد أية معلومات عن نوع البيانات التي تظهر في هذا العنصر.

وفي بعض الأوقات يكون هناك كل من DTD بالإضافة إلى المواصفات المكتوبة. ومن أحـــد أمثلة ذلك هـو Microsoft and Marimba's Open Software Description. غـير أن المشكلة التي تكمن في المواصفات المكتوبة هي أنها تترك بعض المعلومات بغير ذكـــر، فمثـلاً مواصفة OSD لم تذكر كم عدد العناصر التابعة التي تظهر في العنصـــر الأصـل أو فـي أي ترتيب، ولكن DTD توضح ذلك وهو متروك للجزء المكتوب من المواصفة.



يمكن بل ومن المستحسن أن يظهر هذا الجزء في أحد التعليقات فـــي DTD. ولا يستطيع معالج XML أن يتحقق من صحة هذا القيد بمفرده، وهـــو الأمـــر الـــذي يحتاج مستوى أعلى من المعالجة. ومن الممكـــن أن يتـم تبسـيط DTD علـــى المستخدم ببعض التعليقات البسيطة. وفي الوقت الحالي لا يحتـــوي OSD علـــى DTD متصل به.

وكل هذه هي عبارة عن أمثلة لتطبيقات XML العامة. غير أن هناك الكثير مـــن المؤسســات والشركات والوكالات والحكومات التي تستخدم تطبيقات XML الداخليـــة الخاصـــة بـــها والتـــي يستخدمونها في التعامل مع المستندات الخاصة بهم غير أنها قد لا تكون في نفس جودة التطبيقات العامة. وقد تحتاج إلى حفظ تصميم DTD الذي تمت كتابته بواسطة أحد الأشخاص منذ فترة ثـــم تزايد عدد عناصره بمرور الزمن.

ومن الواضح أنه إذا زاد التوثيق الموجود في تطبيق XML، وكلما كانت هذه الوثائق مكتوبـــة بعناية، كلما زادت سهولة تعلمه واستخدامه. غير أنه من سو الحظ أنه دائماً ما يرجأ التفكير فـــي الوثائق.ودائماً ما تكون محتاجاً للعمل مستخدماً DTD. ومن ثم فإنه مطلوب منك أن تقـــرأ DTD وتفهمه جيداً ثم تقوم بكتابة بعض المستندات الاختبارية التي تقوم بعملية التحقق من الصحة وحتى تتيح لك معرفة الأجراءات المتاحة من الإجراءات غير المتاحة.لذلك يجب عليك أيضا أن تقـــوم بتحويل مل تقرؤه إلى أمثلة لحالات الترميز المسموح بها.

ويتم في هذا الفصل استكشاف XHTML من W3C وهو من أحد أفضل DTD الموثقة. وبـــدلاً من أن تقوم بقراءة المواصفة المكتوبة قم بقراءة ملفات DTD الفعلية. وسوف يتم عرض التقنيات المستخدمة التي يمكنك أن تستخدمها في فهم هذه النماذج من DTD، حتى في غيـــاب المواصفـــة المكتوبة.

ما هو XHTML

هو الجهد المبذول من W3C's لإعادة كتابة HTML على شكل XML، مما يتطلب بعض التعديلات على HTML. فعلامات النهاية تكون لازمة للعناصر التي دائماً ما يتم حذفها مثل p وdt، كما أن العناصر الفارغة مثل hr لابد أن تنتهي بالعلامة </ بدلاً من العلامة >. كما يجب أن يتم اقتباس قيم السمة. ويتم وضع كل أسماء وسمات العناصر في حالة الأحرف الصغيرة.

كما يقوم XHTML بتوفير DTD يخص HTML يمكن استخدامه للتحقق من صحة مستندات HTML. وفي الواقع تقوم بتوفير:

- DTD المقيدة التي تخص XHTML لمستندات HTML الجديدة.
- DTD الواسعة التي تخص XHTML لمستندات HTML القديمة المحولة التي مازالت تستخدم العلامات السيئة مثل applet.
- مجموعة إطار XHTML التي تخص DTD للمستندات التي تستخدم الأطر.
- ويمكنك اختيار ما يناسبك منها.

لماذا التأكد من صحة HTML

المستندات الصالحة لا تكون مطلوبة بشكل حتمي في HTML غير أنها تسهل من عمل المستعرضات عند عرضها وفهمها. ويستطيع مستند HTML الصالح أن يقوم بالتنبؤ في عدة مستعرضات غير أن المستند غير الصالح لا يستطيع القيام بمثل هذه المهمة.

وحتى الآن تتنافس المستعرضات على مقدرتها على التعرف على HTML المقطوعة. فيقوم مثلاً Internet Explorer بملأ الفراغات في علامة النهاية </table> بينما لا يقوم Netscape Navigator بمثل ذلك. ومن ثم فإنه هناك الكثير الصفحات على موقع مايكروسوفت "التي تم اختبارها بواسطة Internet Explorer" تحتوي على علامات </table> فارغة ولا يمكن أن يتم عرضها في Netscape Navigator. وعلى كل حال فإن هذا لن يحدث إذا تطلبت مايكروسوفت HTML صالحة على الصفحة الخاصة بها.

ومن الأمور الصعبة القيام باختبار الصفحات في جزء صغير من المستعرضات التي يستخدمها المستخدمون. حتى القيام باختبار آخر الإصدارات من Internet Explorer وNetscape Navigator من أحد الأمور الصعبة.

وضع النسخة التمهيدية من دليل عمل XHTML في وحدات نمطية

يغطي هذا الفصل النسخة التمهيدية من مواصفة دليل العمل Modularized XHTML الخاصة بالسادس من أبريل لعام ١٩٩٩ والتي قد تم تغييرها. وحالة هذا الإصدار كما وضحها W3C's هي كالتالي:

هذا المستند هو دليل العمل الذي يخص W3C's HTML Working Group. وقد يتم تحديثها أو استبدالها أو معالجتها بأحد المستندات الأخرى في أي وقت. ومن غير المناسب أن تقوم باستخدام W3C Working Draft. وهو يعبر عن العمل في حالة تقدمه ولا تعبر عن أنها ملحقاً يخص W3C.

تم إنتاج هذا المستند باعتباره جزءاً من W3C HTML Activity ويتم مناقشة أهداف HTML Working Group "للأعضاء فقط" في مخطط HTML Working Group "للأعضاء فقط".

وفي الوقت الحالي فإن العرض التمهيدي هو من ٦ أبريل ١٩٩٩. ومن الممكن أن تقوم بتحميل هذه النسخة من (http://www.w3.ORG/TR/1999/xhtml-modularization-19990406). ويحتوي هذا المستند على العديد من التفاصيل عن XHTML وعن إعادة كتابة صفحات الويب في XML المتوافق مع HTML. والنسخة الحديثة منه متاحة على الويب في (http://www.w3.ORG/TR/xhtml-modularization). ويركز هذا الفصل على قراءة DTD التي تخص XHTML. والملفات التي يتم مناقشتها فيما يلي تكون معرضة لـ W3C Document Notice، الذي يتم إعادة إنتاجه في الشريط الجانبي.

بنية قراءة تعريفات نوع المستند XHTML

تعد لغة HTML من أحد التطبيقات المعقدة للغة XML. ويمكن أن اختيار لمستندات XHTML أحد DTD التي تم ذكرها فيما سبق. ويتم تقسيم HTML DTD المذكورة هنا إلى ٤٠ ملف وما يزيد على ٢٠٠٠سطر من التعليمات البرمجية. ويتم اتصال هذه الملفات من خلال كينونات المعامل. وتقسيم DTD إلى هذه الملفات يمكن أن يسهل عملية فهمها والأكثر من ذلك أنه يمكن مشاركة الأجزاء المشتركة بين الإصدارات الثلاثة من XHTML DTD وهي المقيدة والواسعة ومجموعة الإطار.

ملاحظة المستند

يتم توفير المستندات على موقع W3C عن طريق المتمتعين بحقوق الطبع. يسمح للمستخدم بالتمتع بكل مميزات الطباعة والنسخ والاستخدام المجاني كما يسمح له بالتعامل مع كل الملحقات الخاصة بهذا البرنامج مثل:

ملاحظة المستند

١- أحد الارتباطات مع URL مع مستند W3C الأصلي.

٢- حق الطبع الخاص بالكاتب الأصلي المتاح مسبقاً.

٣- STATUS الخاصة بالمستند W3C إذا كان هذا متاحاً.

ويجب أن يتم توفير هذا الإعلان كلما أمكن ذلك

غير أنه ليس من حق المستخدم القيام بأي تعديل. في المستند W3C الأصلي.

وإذا كانت المساحة هنا تسمح لنا، لابد أن نقوم بتضمين النص الكامل لـهذه NOTICE. ونحن نطالب بتوفير نسبة الكاتب في أي برامج أو مستندات أو منتجات أخـــرى يتم إنشاؤها وفقاً لتنفيذ محتويات هذا المستند أو أي جزء منه.

وليس هناك حق لإنشاء تعديلات أو اشتقاقات لمستندات يتم ضمانها وفقاً لهذا الإذن.

وقد لا يتم استخدام الاسم والعلامات التجارية لحاملي حق النشر في الإعلان عـــن أو نشر ما يتعلق بهذا المستند أو بمحتوياته بدون إذن مكتوب سابق ومحـــدد. وسـيظل هناك جزء من حق النشر في هذا المستند دائماً لدى حاملي حق النشر.

وفيما يلي الأنواع الثلاثة من DTD التي يمكن استخدامها عن طريق HTML فـــي مســتندات XML:

١- DTD المقيدة التي تخص XHTML للمستندات الجديدة الخاصة بلغة HTML.

٢- DTD واسعة التي تخص XHTML لمستندات HTML القديمة المحولـــة التـــي مـــازالت تستخدم الأنواع غير الجيدة من علامات التبويب مثل applet.

٣- مجموعة إطار DTD التي تخص XHTML للمستندات التي تستخدم الأطر.

وكل من هذه DTD الأساسية له التنسيقات التالية:

١- التعليق بالعنوان وحيز الأسماء وحقوق الطبع والمعرف العـــام الرســمي والمعلومـــات الأخرى التي قد يحتاجها من يستخدم DTD.

٢- تعريفات معاملات الكينونة التي تم مراجعتها والتي تقوم بإلغاء كينونات المعامل التي تـم تعريفها في الوحدات النمطية.

٣- مراجع كينونة المعامل الخارجي لتصدير مجموعات الوحدات النمطية والكينونة.

DTD المقيدة التي تخص XHTML

تعتبر هذه DTD المقيدة التي تخص XHTML (XHTML1-s.dtd) والموضحة في تعليمات البرمجة ٢٠-١ خاصة بمستندات HTML والتي يمكن أن تتوافق مع أكثر المتطلبات صرامة في لغة XML، ومن ثم فإنه ليست هناك حاجة إلى استخدام العناصر القديمة وغير المحبذة من لغة HTML مثل applet وbasefont. وهي لا تعمل مع الأطر كما تقوم بإلغاء كل إمكانيات العمل مع عناصر التقديم مثل font وcenter.

تعليمات البرمجة ٢٠-١ : XHTML1-s.dtd:

```
<!- ................................................................. ->
<!- XHTML 1.0 Strict DTD ............................... ->
<!- file: XHTML1-s.dtd
->

<!-  XHTML 1.0 Strict DTD
    xml:lang="en" lang="en" >
     ...
     </html>
->

<!- The version attribute has historically been a container
    for the DTDحص public identifier (an FPI), but is unused
    in Strict:  ->
<!ENTITY % HTML.version  "" >
<!ENTITY % Version.attrib "" >

<!- The xmlns attribute on <html> identifies the
    default namespace to namespace-aware applications:  ->
<!ENTITY % XHTML.ns  "http://www.w3.org/TR/1999/REC-html-in-xml"
>

<!- reserved for future use with document profiles ->
<!ENTITY % XHTML.profile  "" >

<!- used to ignore Transitional features within modules ->
```

```
<!ENTITY % XHTML.Transitional    "IGNORE" >


<!- XHTML Base Architecture Module (optional) ......... ->
<!ENTITY % XHTML1-arch.module "IGNORE" >
<![%XHTML1-arch.module;[
<!ENTITY % XHTML1-arch.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"
         "XHTML1-arch.mod" >
%XHTML1-arch.mod;
]]>

<!- Common Names Module .............................. ->
<!ENTITY % XHTML1-names.module "INCLUDE" >
<![%XHTML1-names.module;[
<!ENTITY % XHTML1-names.mod
   PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
         "XHTML1-names.mod" >
%XHTML1-names.mod;
]]>

<!- Character Entities Module ........................ ->
<!ENTITY % XHTML1-charent.module "INCLUDE" >
<![%XHTML1-charent.module;[
<!ENTITY % XHTML1-charent.mod
   PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
         "XHTML1-charent.mod" >
%XHTML1-charent.mod;
]]>

<!- Intrinsic Events Module .......................... ->
<!ENTITY % XHTML1-events.module "INCLUDE" >
<![%XHTML1-events.module;[
<!ENTITY % XHTML1-events.mod
   PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
         "XHTML1-events.mod" >
%XHTML1-events.mod;
]]>
```

```
<!- Common Attributes Module .......................... ->
<!ENTITY % XHTML1-attribs.module "INCLUDE" >
<![%XHTML1-attribs.module;[
<!ENTITY % align "" >
<!ENTITY % XHTML1-attribs.mod
    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"
          "XHTML1-attribs.mod" >
%XHTML1-attribs.mod;
]]>

<!- Document Model Module ............................ ->
<!ENTITY % XHTML1-model.module "INCLUDE" >
<![%XHTML1-model.module;[
<!ENTITY % XHTML1-model.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"
          "XHTML1-model.mod" >
%XHTML1-model.mod;
]]>

<!- Inline Structural Module .......................... ->
<!ENTITY % XHTML1-inlstruct.module "INCLUDE" >
<![%XHTML1-inlstruct.module;[
<!ENTITY % XHTML1-inlstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
          "XHTML1-inlstruct.mod" >
%XHTML1-inlstruct.mod;
]]>

<!- Inline Presentational Module ...................... ->
<!ENTITY % XHTML1-inlpres.module "INCLUDE" >
<![%XHTML1-inlpres.module;[
<!ENTITY % XHTML1-inlpres.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
        "XHTML1-inlpres.mod" >
%XHTML1-inlpres.mod;
]]>
```

```
<!- Inline Phrasal Module ............................ ->
<!ENTITY % XHTML1-inlphras.module "INCLUDE" >
<![%XHTML1-inlphras.module;[
<!ENTITY % XHTML1-inlphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
          "XHTML1-inlphras.mod" >
%XHTML1-inlphras.mod;
]]>

<!- Block Structural Module .......................... ->
<!ENTITY % XHTML1-blkstruct.module "INCLUDE" >
<![%XHTML1-blkstruct.module;[
<!ENTITY % XHTML1-blkstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
          "XHTML1-blkstruct.mod" >
%XHTML1-blkstruct.mod;
]]>

<!- Block Presentational Module ...................... ->
<!ENTITY % XHTML1-blkpres.module "INCLUDE" >
<![%XHTML1-blkpres.module;[
<!ENTITY % XHTML1-blkpres.mod
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
          "XHTML1-blkpres.mod" >
%XHTML1-blkpres.mod;
]]>

<!- Block Phrasal Module ............................ ->
<!ENTITY % XHTML1-blkphras.module "INCLUDE" >
<![%XHTML1-blkphras.module;[
<!ENTITY % XHTML1-blkphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
          "XHTML1-blkphras.mod" >
%XHTML1-blkphras.mod;
]]>

<!- Scripting Module ................................ ->
<!ENTITY % XHTML1-script.module "INCLUDE" >
```

```
<![%XHTML1-script.module;[
<!ENTITY % XHTML1-script.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
          "XHTML1-script.mod" >
%XHTML1-script.mod;
]]>

<!- Stylesheets Module ................................ ->
<!ENTITY % XHTML1-style.module "INCLUDE" >
<![%XHTML1-style.module;[
<!ENTITY % XHTML1-style.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"
          "XHTML1-style.mod" >
%XHTML1-style.mod;
]]>

<!- Image Module ...................................... ->
<!ENTITY % XHTML1-image.module "INCLUDE" >
<![%XHTML1-image.module;[
<!ENTITY % XHTML1-image.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
          "XHTML1-image.mod" >
%XHTML1-image.mod;
]]>

<!- Frames Module .................................... ->
<!ENTITY % XHTML1-frames.module  "IGNORE" >
<![%XHTML1-frames.module;[
<!ENTITY % XHTML1-frames.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
          "XHTML1-frames.mod" >
%XHTML1-frames.mod;
]]>

<!- Linking Module .................................... ->
<!ENTITY % XHTML1-linking.module "INCLUDE" >
<![%XHTML1-linking.module;[
<!ENTITY % XHTML1-linking.mod
```

```
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
           "XHTML1-linking.mod" >
%XHTML1-linking.mod;
]]>

<!- Client-side Image Map Module ...................... ->
<!ENTITY % XHTML1-csismap.module "INCLUDE" >
<![%XHTML1-csismap.module;[
<!ENTITY % XHTML1-csismap.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"
         "XHTML1-csismap.mod" >
%XHTML1-csismap.mod;
]]>

<!- Object Element Module ............................ ->
<!ENTITY % XHTML1-object.module "INCLUDE" >
<![%XHTML1-object.module;[
<!ENTITY % XHTML1-object.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
           "XHTML1-object.mod" >
%XHTML1-object.mod;
]]>

<!- Lists Module ..................................... ->
<!ENTITY % XHTML1-list.module "INCLUDE" >
<![%XHTML1-list.module;[
<!ENTITY % XHTML1-list.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
           "XHTML1-list.mod" >
%XHTML1-list.mod;
]]>

<!- Forms Module ..................................... ->
<!ENTITY % XHTML1-form.module "INCLUDE" >
<![%XHTML1-form.module;[
<!ENTITY % XHTML1-form.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
           "XHTML1-form.mod" >
```

وهي في غاية الأهمية إذا كنت تريد أن تنشر أي DTD. ويجب أن يتاح لمن هو خارج نطاق المنظمة نسخ DTD أو وضعها في خدمات الويب أو إرسـالها إلـى مسـتخدمين آخريـن مـع المستندات الخاصة بهم أو عمل الكثير من الأعمال التي قد تكون غير متاحة مع حقـوق الطبـع الأخرى.


```
%XHTML1-form.mod;
]]>

<!- Tables Module ..................................... ->
<!ENTITY % XHTML1-table.module "INCLUDE" >
<![%XHTML1-table.module;[
<!ENTITY % XHTML1-table.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
          "XHTML1-table.mod" >
%XHTML1-table.mod;
]]>

<!- Document Metainformation Module .................. ->
<!ENTITY % XHTML1-meta.module "INCLUDE" >
<![%XHTML1-meta.module;[
<!ENTITY % XHTML1-meta.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
          "XHTML1-meta.mod" >
%XHTML1-meta.mod;
]]>

<!- Document Structure Module ......................... ->
<!ENTITY % XHTML1-struct.module "INCLUDE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
          "XHTML1-struct.mod" >
%XHTML1-struct.mod;
]]>

<!- end of XHTML 1.0 Strict DTD  ......................... ->
<!- ..................................................... ->
```

ويبدأ الملف بأحد التعليقات التي توضح نوع الملف وعبارة أساسية توضح حقوق الطبع.

وهي في غاية الأهمية إذا كنت تريد أن تنشر أي DTD. ويجب أن يتاح لمن هو خارج نطاق المنظمة نسخ DTD أو وضعها في خدمات الويب أو إرسـالها إلـى مسـتخدمين آخريـن مـع المستندات الخاصة بهم أو عمل الكثير من الأعمال التي قد تكون غير متاحة مع حقـوق الطبـع الأخرى.

ثم يأتي بعد ذلك أحد التعليقات التي تحتوي على معلومات عن كيفية استخدام DTD بما في ذلك المعرف الرسمي العام. كما أنه أيضاً يتم توفير حيز الأسماء المفضل بالإضافة إلى أحد الأمثلة التي توضح كيفية استخدام DTD مما له عظيم الفائدة بالنسبة للكاتب.

المرجع يتم مناقشة المعرف العام الرسمي في الفصل الثامن.

ثم يأتي بعد ذلك تعريفات متعددة للكينونات أغلبها يخص التوافق مع الإصدارات الجديدة أو القديمة الخاصة بهذا DTD. وفي النهاية يتعرض إلى تعريفات كينونات المعامل الخارجيDTD ٢٤ والمراجع التي تصدر الوحدات النمطية المستخدمة لعمل DTD الكاملة. وها هي الأخيرة في الملف:

```
<!" Document Structure Module ......................... ">
<!ENTITY % XHTML1-struct.module "INCLUDE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
          "XHTML1-struct.mod" >
%XHTML1-struct.mod;
]]>
```

وتنطبق البنية الأساسية على كل واحد من ٢٤.

١- أحد التعليقات الذي يحدد الوحدة النمطية المراد استيرادها.

٢- تعريف كينونة المعامل الذي يكون اسمه هو نفس اسم الوحدة النمطية التي يتم استيرادها مع تضمين كلمة module في نهايتها والذي يكون النص الذي يحل محله هو INCLUDE أو IGNORE.

٣- يتم تحديد ما إذا كانت IGNORE أم INCLUDE عن طريق قيمة مرجع معامل كينونة المعامل في الخطوة السابقة.

٤- تعريف كينونة معامل خارجي للوحدة النمطية المراد استيرادها مع تذييلها بالكلمة mod يليها مرجع كينونة المعامل الخارجي الذي يقوم بالفعل باستيراد الوحدة النمطية.

وبعد إزالة الأجزاء الخاصة بالوحدة النمطية، تبدو البنية على هذا النحو:

```
<!- Module Name Module ......................... ->
<!ENTITY % XHTML1-module_abbreviation.module "INCLUDE" >
<![%XHTML1-module_abbreviation.module;[
```

```
<!ENTITY % XHTML1-module_abbreviation.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Module Name//EN"
        "XHTML1-module_abbreviation.mod" >
%XHTML1-module_abbreviation.mod;
]]>
```

ويعد تغيير الطريقة التي يتم استيعابها بها أمراً في غاية السهولة، سواء إذا تم تحميل وحدة نمطية معينة عن طريق تغيير قيمة كينونة أحد المعاملات الداخلية من INCLUDE إلى IGNORE أو العكس. وتعمل كينونات المعامل على هيئة مفاتيح تقوم بإغلاق التعريفات المعينة أو بفتحها.

تعريفات نوع المستند الانتقالية للغة XHTML

تتناسب تعريفات نوع المستند الانتقالية للغة XHTML "XHTML1-t.dtd"، كما تسمى أيضاً DTD الواسعة.وهي موضحة في تعليمات البرمجة ٢٠-٢ مستندات HTML التي لم تقوم بالإنتقال إلى HTML 4.0. وتعتمد مثل هذه المستندات على العناصر غير الجيدة مثل applet وcenter. كما أنها تضيف المزيد من الدعم إلى خصائص العرض التقديمي مثل اللون ونمط الرمز النقطي الخاصة بأجزاء القائمة التي تم إحلالها بأوراق النمط في HTML 4.0 المقيدة.

تعليمات البرمجة ٢٠-٢: XHTML1-t.dtd تعريفات نوع المستند الانتقالية للغة XHTML

```
<!- ................................................... ->
<!- XHTML 1.0 Transitional DTD ......................................... ->
<!- file: XHTML1-t.dtd
->

<!-  XHTML 1.0 Transitional DTD

     This is XHTML 1.0, an XML reformulation of HTML 4.0.

     Copyright 1998-1999 World Wide Web Consortium
        (Massachusetts Institute of Technology, Institut
        National de Recherche en Informatique et en
        Automatique, Keio University). All Rights Reserved.
```

Permission to use, copy, modify and distribute the XHTML 1.0 DTD and its accompanying documentation for any purpose and without fee is hereby granted in perpetuity, provided that the above copyright notice and this paragraph appear in all copies. The copyright holders make no representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Murray M. Altheim <altheim@eng.sun.com>
Revision: @(#)XHTML1-t.dtd 1.14 99/04/01 SMI

The XHTML 1.0 DTD is an XML variant based on the W3C HTML 4.0 DTD:

Draft: $Date: 1999/04/02 14:27:27 $

Authors: Dave Raggett <dsr@w3.org>
Arnaud Le Hors <lehors@w3.org>
Ian Jacobs <ij@w3.org>

->

<!- This is the driver file for version 1.0 of the XHTML Transitional DTD.

Please use this formal public identifier to identify it:

"-//W3C//DTD XHTML 1.0 Transitional//EN"

Please use this URI to identify the default namespace:

"http://www.w3.org/TR/1999/REC-html-in-xml"

For example, if you are using XHTML 1.0 directly, use the FPI in the DOCTYPE declaration, with the xmlns attribute on the



document element to identify the default namespace:

<?xml version="1.0" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"XHTML1-t.dtd" >
<html xmlns="http://www.w3.org/TR/1999/REC-html-in-xml"
xml:lang="en" lang="en" >
...
</html>
->

<!- The version attribute has historically been a container
for the DTD's public identifier (an FPI): ->
<!ENTITY % HTML.version "-//W3C//DTD XHTML 1.0 Transitional//EN"
>

<!- The xmlns attribute on <html> identifies the
default namespace to namespace-aware applications: ->
<!ENTITY % XHTML.ns "http://www.w3.org/TR/1999/REC-html-in-xml"
>

<!- reserved for future use with document profiles ->
<!ENTITY % XHTML.profile "" >

<!ENTITY % XHTML1-frames.module "IGNORE" >
<!ENTITY % XHTML.Transitional "INCLUDE" >

<!- XHTML Base Architecture Module (optional) .... ->
<!ENTITY % XHTML1-arch.module "IGNORE" >
<![%XHTML1-arch.module;[
<!ENTITY % XHTML1-arch.mod
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"
"XHTML1-arch.mod" >
%XHTML1-arch.mod;
]]>

<!- Common Names Module .... ->
<!ENTITY % XHTML1-names.module "INCLUDE" >

```
<![%XHTML1-names.module;[
<!ENTITY % XHTML1-names.mod
    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
         "XHTML1-names.mod" >
%XHTML1-names.mod;
]]>

<!- Character Entities Module ........................ ->
<!ENTITY % XHTML1-charent.module "INCLUDE" >
<![%XHTML1-charent.module;[
<!ENTITY % XHTML1-charent.mod
PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
         "XHTML1-charent.mod" >
%XHTML1-charent.mod;
]]>

<!- Intrinsic Events Module .......................... ->
<!ENTITY % XHTML1-events.module "INCLUDE" >
<![%XHTML1-events.module;[
<!ENTITY % XHTML1-events.mod
    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
         "XHTML1-events.mod" >
%XHTML1-events.mod;
]]>

<!- Transitional Attributes Module .................... ->
<!ENTITY % XHTML1-attribs-t.module "INCLUDE" >
<![%XHTML1-attribs-t.module;[
<!ENTITY % XHTML1-attribs-t.mod
PUBLIC "-//W3C//ENTITIES XHTML 1.0 Transitional Attributes//EN"
      "XHTML1-attribs-t.mod" >
%XHTML1-attribs-t.mod;
]]>

<!- Transitional Document Model Module ............... ->
<!ENTITY % XHTML1-model-t.module "INCLUDE" >
<![%XHTML1-model-t.module;[
<!ENTITY % XHTML1-model-t.mod
```

```
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Document
Model//EN"
          "XHTML1-model-t.mod" >
%XHTML1-model-t.mod;
]]>

<!- Inline Structural Module ........................ ->
<!ENTITY % XHTML1-inlstruct.module "INCLUDE" >
<![%XHTML1-inlstruct.module;[
<!ENTITY % XHTML1-inlstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
          "XHTML1-inlstruct.mod" >
%XHTML1-inlstruct.mod;
]]>

<!- Inline Presentational Module ..................... ->
<!ENTITY % XHTML1-inlpres.module "INCLUDE" >
<![%XHTML1-inlpres.module;[
<!ENTITY % XHTML1-inlpres.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
  "XHTML1-inlpres.mod" >
%XHTML1-inlpres.mod;
]]>

<!- Inline Phrasal Module ............................ ->
<!ENTITY % XHTML1-inlphras.module "INCLUDE" >
<![%XHTML1-inlphras.module;[
<!ENTITY % XHTML1-inlphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
          "XHTML1-inlphras.mod" >
%XHTML1-inlphras.mod;
]]>

<!- Block Structural Module .......................... ->
<!ENTITY % XHTML1-blkstruct.module "INCLUDE" >
<![%XHTML1-blkstruct.module;[
<!ENTITY % XHTML1-blkstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
```

```
        "XHTML1-blkstruct.mod" >
%XHTML1-blkstruct.mod;
]]>

<!- Block Presentational Module ...................... ->
<!ENTITY % XHTML1-blkpres.module "INCLUDE" >
<![%XHTML1-blkpres.module;[
<!ENTITY % XHTML1-blkpres.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
        "XHTML1-blkpres.mod" >
%XHTML1-blkpres.mod;
]]>

<!- Block Phrasal Module ............................ ->
<!ENTITY % XHTML1-blkphras.module "INCLUDE" >
<![%XHTML1-blkphras.module;[
<!ENTITY % XHTML1-blkphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
          "XHTML1-blkphras.mod" >
%XHTML1-blkphras.mod;
]]>

<!- Scripting Module ................................ ->
<!ENTITY % XHTML1-script.module "INCLUDE" >
<![%XHTML1-script.module;[
<!ENTITY % XHTML1-script.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
          "XHTML1-script.mod" >
%XHTML1-script.mod;
]]>

<!- Stylesheets Module ............................... ->
<!ENTITY % XHTML1-style.module "INCLUDE" >
<![%XHTML1-style.module;[
<!ENTITY % XHTML1-style.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"
          "XHTML1-style.mod" >
%XHTML1-style.mod;
```

```
]]>

<!- Image Module ..................................... ->
<!ENTITY % XHTML1-image.module "INCLUDE" >
<![%XHTML1-image.module;[
<!ENTITY % XHTML1-image.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
          "XHTML1-image.mod" >
%XHTML1-image.mod;
]]>

<!- Frames Module .................................... ->
<![%XHTML1-frames.module;[
<!ENTITY % XHTML1-frames.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
          "XHTML1-frames.mod" >
%XHTML1-frames.mod;
]]>

<!- Linking Module .................................... ->
<!ENTITY % XHTML1-linking.module "INCLUDE" >
<![%XHTML1-linking.module;[
<!ENTITY % XHTML1-linking.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
          "XHTML1-linking.mod" >
%XHTML1-linking.mod;
]]>

<!- Client-side Image Map Module ...................... ->
<!ENTITY % XHTML1-csismap.module "INCLUDE" >
<![%XHTML1-csismap.module;[
<!ENTITY % XHTML1-csismap.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"
        "XHTML1-csismap.mod" >
%XHTML1-csismap.mod;
]]>

<!- Object Element Module ............................ ->
```

```
<!ENTITY % XHTML1-object.module "INCLUDE" >
<![%XHTML1-object.module;[
<!ENTITY % XHTML1-object.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
        "XHTML1-object.mod" >
%XHTML1-object.mod;
]]>

<!- Java Applet Element Module ........................ ->
<!ENTITY % XHTML1-applet.module "INCLUDE" >
<![%XHTML1-applet.module;[
<!ENTITY % XHTML1-applet.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Java Applets//EN"
        "XHTML1-applet.mod" >
%XHTML1-applet.mod;
]]>

<!- Lists Module ..................................... ->
<!ENTITY % XHTML1-list.module "INCLUDE" >
<![%XHTML1-list.module;[
<!ENTITY % XHTML1-list.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
        "XHTML1-list.mod" >
%XHTML1-list.mod;
]]>

<!- Forms Module ..................................... ->
<!ENTITY % XHTML1-form.module "INCLUDE" >
<![%XHTML1-form.module;[
<!ENTITY % XHTML1-form.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
        "XHTML1-form.mod" >
%XHTML1-form.mod;
]]>

<!- Tables Module .................................... ->
<!ENTITY % XHTML1-table.module "INCLUDE" >
<![%XHTML1-table.module;[
```

```
<!ENTITY % XHTML1-table.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
        "XHTML1-table.mod" >
%XHTML1-table.mod;
]]>

<!- Document Metainformation Module .................. ->
<!ENTITY % XHTML1-meta.module "INCLUDE" >
<![%XHTML1-meta.module;[
<!ENTITY % XHTML1-meta.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
        "XHTML1-meta.mod" >
%XHTML1-meta.mod;
]]>

<!- Document Structure Module ......................... ->
<!ENTITY % XHTML1-struct.module "INCLUDE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
        "XHTML1-struct.mod" >
%XHTML1-struct.mod;
]]>

<!- end of XHTML 1.0 Transitional DTD ................................... ->
<!- ................................................... ->
```

تم تنظيم DTD على نفس طريقة الأسطر كما هو الحال مع DTD المقيدة. فتخبرك التعليقات بما يجب أن تتبعه لاستخدام DTD. ثم تأتي بعد ذلك تعريفات الكينونات والتي هي هامة بالنسبة للوحدات النمطية التي يتم استيرادها وخاصة HTML.TRANSITIONAL والتي يتم تعريفها هنا على هيئة INCLUDE. كما تم تعريفها على شكل IGNORE في DTD المقيد. ومن ثم فإن الوحدات النمطية يمكن أن تستخدم ذلك لتوفير بعض الملامح التي يتم تطبيقها عندما يتم استخدام DTD الإنتقالية فقط. وغي النهاية يتم استيراد الوحدات النمطية المتعددة. ويكمن الفرق بين DTD الانتقالية والمقيدة في أي الوحدات النمطية يتم استيرادها وكيف يتم الغاء معاملات الكينونات. ويدعم DTD الانتقالي مجموعة من DTD المقيدة.

مجموعة إطار تعريفات نوع المستند للغة XHTML

تعتبر مجموعة إطار تعريفات نوع المستند للغة XHTML الموضحة في تعليمات البرمجة ٢٠-٣ هي مجموعة إضافية من DTD الانتقالية التي تدعم الإطار.

تعليمات البرمجة ٢٠-٣: XHTML1-f.dtd: تعريفات نوع المستند الواسع Voyager للمستندات ذات الأطر

```
<!- ................................................. ->
<!- XHTML 1.0 Frameset DTD ........................................... ->
<!- file: XHTMl1-f.dtd
->

<!- XHTML 1.0 Frameset DTD

    This is XHTML 1.0, an XML reformulation of HTML 4.0.

    Copyright 1998-1999 World Wide Web Consortium
      (Massachusetts Institute of Technology, Institut
      National de Recherche en Informatique et en
      Automatique, Keio University). All Rights Reserved.

    Permission to use, copy, modify and distribute the XHTML
    1.0 DTD and its accompanying documentation for any
    purpose and without fee is hereby granted in perpetuity,
    provided that the above copyright notice and this
    paragraph appear in all copies. The copyright holders
    make no representation about the suitability of the DTD
    for any purpose.

    It is provided "as is" without expressed or implied
    warranty.

      Author:    Murray M. Altheim <altheim@eng.sun.com>
      Revision:  @(#)XHTML1-f.dtd 1.17 99/04/01 SMI
```

```
     The XHTML 1.0 DTD is an XML variant based on
     the W3C HTML 4.0 DTD:

       Draft:      $Date: 1999/04/02 14:27:26 $

       Authors:    Dave Raggett <dsr@w3.org>
                   Arnaud Le Hors <lehors@w3.org>
                   Ian Jacobs <ij@w3.org>

->
<!-  This is the driver file for version 1.0 of
     the XHTML Frameset DTD.

     Please use this formal public identifier to identify it:

        "-//W3C//DTD XHTML 1.0 Frameset//EN"

     Please use this URI to identify the default namespace:

        "http://www.w3.org/TR/1999/REC-html-in-xml"

     For example, if you are using XHTML 1.0 directly, use the
     FPI in the DOCTYPE declaration, with the xmlns attribute
     on the document element to identify the default
     namespace:

  <?xml version="1.0" ?>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
                  "XHTML1-f.dtd" >
  <html xmlns="http://www.w3.org/TR/1999/REC-html-in-xml"
       xml:lang="en" lang="en" >
  ...
  </html>
->

<!- The version attribute has historically been a container
    for the DTD's public identifier (an FPI): ->
<!ENTITY % HTML.version  "-//W3C//DTD XHTML 1.0 Frameset//EN" >
```

```
<!- The xmlns attribute on <html> identifies the
    default namespace to namespace-aware applications: ->
<!ENTITY % XHTML.ns
  "http://www.w3.org/TR/1999/REC-html-in-xml" >

<!- reserved for future use with document profiles ->
<!ENTITY % XHTML.profile  "" >

<!ENTITY % XHTML1-frames.module  "INCLUDE" >
<!ENTITY % XHTML.Transitional    "INCLUDE" >


<!- declare and instantiate the XHTML Transitional DTD  ->
<!ENTITY % XHTML1-t.dtd
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"XHTML1-t.dtd" >
%XHTML1-t.dtd;

<!- end of XHTML 1.0 Frameset DTD ...................................... ->
<!- .................................................
```

وهذه DTD قد تم تنظيمها على نحو مختلف غير DTD التي تم عرضها فيما سبق. وبدلاً من إعادة كل التعريفات التي تم ذكرها سابقاً في DTD الانتقالية وهي ببساطة تقوم باستيراد DTD باستخدام كينونة المعامل الخارجي XHTML1-t.dtd. وقبل عمل ذلك تقوم بتعريف XHTML1-frames.module على شكل INCLUDE. كا تم تعريف هذه الكينونة في DTDالانتقالية على هيئة IGNORE. غير أن التعريف الموجود هنا يأتي في مرحلة متقدمة. وتغير هذه DTD نفس DTD التي تقوم باستيرادها.

ومن الممكن أن تقوم بعمل DTD مقيدة تستخدم الأطر عن طريق استيراد DTD المقيدة بدلاً من DTD الانتقالية كما يلي:

```
<!-- declare and instantiate the XHTML Strict DTD  ->
<!ENTITY % XHTML1-s.dtd
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
         "XHTML1-s.dtd" >
%XHTML1-s.dtd;
```

غير أن DTD الأخرى مثل XHTML1-s.dtd وXHTML1-t.dtd وXHTML1-t.dtd هي الأنواع الثلاثة للمستندات الأساسية التي يمكن إنشاؤها باستخدام XHTML. أحدها هو المستند XHTML1-m.dtd وهو عبارة عن DTD يشتمل على كل من HTML وMathML "مع وجود اختلافين حتى تستطيع لغة MathML أن تتوافق تماماً مع HTML".

كما انه هناك ثلاثة إصدارات أخرى غير مفصلة تخص DTD التي تستخدم ملف مفرد DTD بدلاً من عدة وحدات نمطية منفصلة. وهي لا تقوم بتعريف تطبيقات XML كما نها ليست بنفس سهولة DTD التي تم مناقشتها هنا.غير أنها أكثر سهولة ففي وضعها على صفحة الويب. وهي تتضمن:

- XHTML1-s-flat.dtd: وهي عبارة عن XHTML DTD مقيدة في ملف واحد.
- XHTML1-t-flat.dtd: وهي عبارة عن XHTML DTD انتقالية في ملف واحد.
- XHTML1-f-flat.dtd: وهي عبارة عن XHTML DTD انتقالية مع دعم من الإطار في ملف واحد.

ومن الممكن أن تقوم بإنشاء DTD الخاص بك والذي يقوم بمقارنة الأجزاء مع HTML القياسي. كما يمكنك أن تختار ما تريده وتترك ما لا تحتاج إليه.كما أنه يمكنك أيضاً أن تقوم بخلطها مع DTD أخرى . غير أنه قبل أن تقوم بذلك، فإنك تحتاج إلى نظرة أكثر دقة على الوحدات النمطية المتاحة.

الوحدات النمطية الخاصة بلغة XHTML

تقوم لغة XHTML بتقسيم HTML إلى ٢٨ وحدة نمطية مختلفة. وكل وحدة منها هي DTD لمجموعة فرعية من عناصر HTML. كما أنه يمكن أن تستخدم على نحو منفصل عن بقية الوحدات. فيمكنك مثلاً أن تضيف دعم أساسي للجدول إلى تطبيق XML الخاص بك. عن طريق استيراد الوحدات النمطية الخاصة بالجدول إلى DTD وتوفير تعريفات إلى بعض كينونات المعاملات مثل Inline وFlow التي تشتمل على عناصر خاصة بالحصيلة اللغوية. وتتضمن الوحدات النمطية المتاحة:

١- XHTML1-applet.mod
٢- XHTML1-arch.mod
٣- XHTML1-attribs-t.mod
٤- 4.XHTML1-attribs.mod
٥- XHTML1-blkphras.mod

٦- XHTML1-blkpres.mod

٧- XHTML1-blkstruct.mod

٨- XHTML1-charent.mod

٩- XHTML1-csismap.mod

١٠- XHTML1-events.mod

١١- XHTML1-form.mod

١٢- XHTML1-frames.mod

١٣- XHTML1-image.mod

١٤- XHTML1-inlphras.mod.

١٥- XHTML1-inlpres.mod.

١٦- XHTML1-inlstruct.mod.

١٧- XHTML1-linking.mod.

١٨- XHTML1-list.mod

١٩- XHTML1-tables.mod

٢٠- XHTML1-meta.mod

٢١- XHTML1-model-t.mod

٢٢- XHTML1-model.mod.

٢٣- XHTML1-names.mod.

٢٤- XHTML1-object.mod.

٢٥- XHTML1-script.mod.

٢٦- XHTML1-struct.mod.

٢٧- XHTML1-style.mod

٢٨- XHTML1-table.mod

وتستخدم مجموعة الإطار كل الوحدات النمطية البالغ عددها ٢٨. كما تستخدم DTD الانتقالية أغلبها فيما عدا XHTML1-frames وXHTML1-arch وXHTML1-attribs و-XHTML1-model. كما تستخدم DTD المقيدة ٢٢ فقط، حيث تحذف XHTML1-arch و-XHTML1-attribs–t وXHTML1-frames وHTML1-applet وXHTML1-model-t.

وحدات النمطية للأسماء الشائعة

ما يتم استيراده من الكينونات من الثلاثة وحدات النمطية الأولى هــي XHTML1-names.mod، وهي أسماء الوحدات النمطية الشائعة، كما هو موضح في تعليمات البرمجة.

تعليمات البرمجة ٢٠-٤: XHTML1-names.mod: الوحدة النمطية XHTML والتي تعرف الأسماء شائعة الاستخدام

```
<!- .................................................. ->
<!- XHTML 1.0 Document Common Names Module ............. ->
<!- file: XHTML1-names.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-names.mod 1.16 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    Identifiers:

    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
    SYSTEM "XHTML1-names.mod"

    Revisions:
# 1999-01-31  added URIs PE for multiple URI attribute values
    .................................................. ->

<!- i. Common Names

    defines the following common names, many of these imported
    from other specifications and standards.
->

<!- .... Imported Names .... ->

<!- media type, as per [RFC2045] ->
<!ENTITY % ContentType "CDATA" >
```

```
<!- comma-separated list of media types, as per [RFC2045] ->
<!ENTITY % ContentTypes "CDATA" >

<!- a character encoding, as per [RFC2045] ->
<!ENTITY % Charset "CDATA" >

<!- a space separated list of character encodings,
   as per [RFC2045] ->
<!ENTITY % Charsets "CDATA" >

<!- date and time information. ISO date format ->
<!ENTITY % Datetime "CDATA" >

<!- a single character from [ISO10646] ->
<!ENTITY % Character "CDATA" >

<!- a language code, as per [RFC1766] ->
<!ENTITY % LanguageCode "NMTOKEN" >

<!- space-separated list of link types ->
<!ENTITY % LinkTypes "NMTOKENS" >

<!- single or comma-separated list of media descriptors ->
<!ENTITY % MediaDesc "CDATA" >

<!- one or more digits (NUMBER) ->
<!ENTITY % Number "CDATA" >

<!- a Uniform Resource Identifier, see [URI] ->
<!ENTITY % URI "CDATA" >

<!- a space-separated list of Uniform Resource Identifiers, see [URI] ->
<!ENTITY % URIs "CDATA" >

<!- script expression ->
<!ENTITY % Script "CDATA" >
```

```
<!- style sheet data ->
<!ENTITY % StyleSheet "CDATA" >

<!ENTITY % Text "CDATA" >

<!-Length defined in strict DTD for cellpadding/cellspacing->

<!- nn for pixels or nn% for percentage length ->
<!ENTITY % Length "CDATA" >

<!- pixel, percentage, or relative ->
<!ENTITY % MultiLength "CDATA" >

<!- comma-separated list of MultiLength ->
<!ENTITY % MultiLengths "CDATA" >

<!- integer representing length in pixels ->
<!ENTITY % Pixels "CDATA" >

<!- render in this frame ->
<!ENTITY % FrameTarget "CDATA" >

<!- a color using sRGB: #RRGGBB as Hex values ->
<!ENTITY % Color "CDATA" >

<!- end of XHTML1-names.mod ->
```

ومن غير السهل فهم مثل هذه DTD لذلك يجب عليك أن ترتبها بطريقة أكثر سهولة. ويقسم الجدول ٢٠-١ قسم Imported Names إلى ثلاثة أعمدة وهي اسم كينونة المعامل وقيمة كينونة المعامل والتعليق المرتبط بكل كينونة معامل. ويوضح شكل هذا الجدول أن المسؤولية الأساسية لهذه الوحدات النمطية هي توفير كينونات للمعاملات لاستخدامه على هيئة نماذج لمحتوى العنصر.

الجدول ٢٠-١

ملخص جزء الاسماء التي تم استيرادها

اسم كينونة المعامل	قيمة كينونة المعامل	التعليق المتعلق بكينونة المعامل
ContentType	CDATA	نوع Media مثل per[RFC2045]
ContentTypes	CDATA	قائمة مفصولة بفاصلة من أنواع MEDIA مثل per [RFC2045]
Charset	CDATA	رموز للحروف مثل [RFC2045]
Charsets	CDATA	قائمة مفصولة بمسافة من رموز الحروف مثل [RFC2045]
Datetime	CDATA	وتذكر زمن ووقت المعلومات. تنسيق تاريخ ISO.
Character	CDATA	حرف واحد من حرف واحد من [ISO10646]
LanguageCode	CDATA	رمز لغة مثل [RFC1766]
LinkTypes	NMTOKENS	قائمة مفصولة بمسافة من أنواع الارتباطات.
MediaDesc	CDATA	قائمة واحدة أو مفصولة بفاصلة من واصفات MEDIA
Number	CDATA	خانة أو أكثر(NUMBER)
URI	CDATA	A Uniform Resource Identifier راجع URI
URIS	CDATA	قائمة مفصولة بمسافة من A Uniform Resource Identifier
Script	CDATA	تعبير Script
StyleSheet	CDATA	بيانات ورقة النمط

الجدول ٢٠-١

ملخص جزء الأسماء التي تم استيرادها

اسم كينونة المعامل	قيمة كينونة المعامل	التعليق المتعلق بكينونة المعامل
Text	CDATA	
Legnth	CDATA	Nn خاص بالبكسل أو nn% لنسبة الطول.
MultiLegnth	CDATA	بكسل أو نسبة أو نسبي
MultiLegnths	CDATA	قائمة مفصولة بمسافة منMultiLength
Pixels	CDATA	عدد صحيح يمثل الطول بالبكسل.
FrameTarget	CDATA	الأداء في هذا الإطار
Color	CDATA	لون يستخدم sRGB: #RRGGBB as Hex values

ومما هو لافت للنظر في هذا الجدول هو عدد مترادفات CDATA. حيث أن كل كينونات المعاملات ما هي إلا مترادفات مختلفة بالنسبة إلى CDATA ما عدا واحد منها فقط. (Heading.class) (Heading.class) (Heading.class) (Heading.class) إلى ذلك فإن كل كاتب من كتاب DTD قد يحتاج إلى أن تمكنه إحدى اللغات التي تمكنه من فرض متطلبات أكثر تفصيلاً على قيم الخصائص.وربما يمكن في المستقبل أن تستطيع أن تقوم بكتابة التعريفات بهذه الطريقة:

```
<!ATTLIST img
  src        URI        #REQUIRED
  alt        String     #REQUIRED
  longdesc   URI        #IMPLIED
  height     Integer    #IMPLIED
  width      Integer    #IMPLIED
  usemap     URI        #IMPLIED
  ismap      (ismap)    #IMPLIED
  author     CDATA      #IMPLIED
  copyright  CDATA      #IMPLIED
  >
```

في هذه الحالة يمكن للكاتب بدلاً من إيجاد واستبدال كل المواقع في DTD حيث تستخدم CDATA كطول أو سلسلة أو URI أو عدد صحيح تغير تعريف %Length و%URI و%Text ومراجع الكينونة مثل ما يلي:

```
<!ENTITY % Length "integer">
<!ENTITY % URI    "URI">
<!ENTITY % Text   "string">
```

في الواقع العملي لن $ يبدو أي أسلوب انتهاء لسمات كتابة البيانات في XML مثل المثال السابق ولكن سيمكن دمجها في XHTML DTD بسرعة بواسطة ضبط بعض تعريفات الكينونات في مستند DTD الرئيسي دون محاولة تحرير ٢٨ وحدة نمطية.

الوحدات النمطية لكينونات الأحرف

توضح تعليمات برمجة ٢٠-٥ الوحدة النمطية الثانية التي تقوم DTD الثلاثة باستيرادها وهي XHTML1-charent.mod وتستورد تلك الوحدة النمطية DTD الذي يعرف مجموعات الكينونات لكينونات HTML القياسي مثل   وα للأحرف التي توجد صعوبة في كتابتها وتلك المجموعات هي:

- XHTML1-lat1.ent، الأحرف ١٦٠ إلى٢٥٥ من Latin-1 وتوضحها تعليمات البرمجة ٢٠-٣٠.
- XHTML1-symbol.ent ينسق ويصنف الأحرف وعلامات الترقيم المفيدة التي لا تكون مدرجة في مجموعة Latin-1 مثل علامة Euro والواصلة الطويلة وتوضيحها تعليمات برمجة ٢٠-٣١.
- XHTML1-special.ent، الهجاء اليوناني وتصنيف الرموز الشائعة الاستخدام في الرياضيات مثل ∞ و∫ وتوضيحها تعليمات برمجة ٢٠-٣٢.

تعليمات البرمجة ٢٠-٥: XHTML1-charent.mod وحدة XHTML النمطية التي تعرف الكينونات شائعة الاستخدام

```
<!- ..................................................... ->
<!- XHTML 1.0 Character Entities Module ................................ ->
<!- file: XHTML1-charent.mod
This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-charent.mod 1.16 99/04/01
```

```
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
    SYSTEM "XHTML1-charent.mod"

    Revisions: (none)
    ................................................ ->

<!- v. Character Entities for XHTML

    declares the set of character entities for XHTML,
    including Latin 1, symbol and special characters.
->

<!- to exclude character entity declarations from a normalized
    DTD, declare %XHTML1.ents; as "IGNORE" in the internal
    subset of the dummy XHTML file used for normalization.
->
<!ENTITY % XHTML1.ents "INCLUDE" >

<![%XHTML1.ents;[
<!ENTITY % XHTML1-lat1
   PUBLIC "-//W3C//ENTITIES Latin 1//EN//XML"
        "XHTML1-lat1.ent" >
%XHTML1-lat1;

<!ENTITY % XHTML1-symbol
   PUBLIC "-//W3C//ENTITIES Symbols//EN//XML"
        "XHTML1-symbol.ent" >
%XHTML1-symbol;

<!ENTITY % XHTML1-special
   PUBLIC "-//W3C//ENTITIES Special//EN//XML"
        "XHTML1-special.ent" >
%XHTML1-special;
```

```
]]>
<!- end of XHTML1-charent.mod ->
```

Notice that a PUBLIC ID tries to load these entity sets. In this case, the public ID may simply be understood by a Web browser as referring to its standard HTML entity set. If not, then the relative URL giving the name of the entity set can find the necessary declarations.

الوحدات النمطية للأحداث الحقيقية

الوحدة النمطية الثالثة التي تستوردها DTD الثلاثة هي الوحدات النمطيـــة للأحـــداث الحقيقيـــة وتعرف تلك الوحدة النمطية السمات الأحداث المختلفة التي قد تقع لعنـــاصر مختلفـــة والممكـــن كتابتها بواسطة استخدام JavaScript وتعريف تلك الوحدة كلا من مجموعة أحداث عامة والتـــي ستستخدم لمعظم العناصر "كينونة Events.attrib" وسمات الإحداث الأكثر تحديداً أي التي تحدد عناصر معينة مثل button وlabel وinput.

تعليمات البرمجة ٢٠-٦: XHTML1-events.mod وحدة الأحداث الحقيقية النمطية

```
<!- ...................................................... ->
<!- XHTML 1.0 Intrinsic Events Module .................. ->
<!- file: XHTML1-events.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-events.mod 1.16 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    Identifiers:

    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
    SYSTEM "XHTML1-events.mod"

    Revisions:
#1999-01-14 transferred onfocus and onblur ATTLIST for 'a' from link
module
```

```
#1999-04-01  transferred remaining events attributes from other
modules
    ........................................................ ->

<!- iv. Intrinsic Event Attributes

    These are the event attributes defined in HTML 4.0,
    Section 18.2.3 "Intrinsic Events"

   "Note: Authors of HTML documents are advised that changes
    are likely to occur in the realm of intrinsic events
    (e.g., how scripts are bound to events). Research in this
    realm is carried on by members of the W3C Document Object
    Model Working Group (see the W3C Web site at
    http://www.w3.org/ for more information)."
->

<!ENTITY % Events.attrib
    "onclick      %Script;          #IMPLIED
     ondblclick   %Script;          #IMPLIED
     onmousedown  %Script;          #IMPLIED
     onmouseup    %Script;          #IMPLIED
     onmouseover  %Script;          #IMPLIED
     onmousemove  %Script;          #IMPLIED
     onmouseout   %Script;          #IMPLIED
     onkeypress   %Script;          #IMPLIED
     onkeydown    %Script;          #IMPLIED
     onkeyup      %Script;          #IMPLIED"
>

<!- additional attributes on anchor element ->

<!ATTLIST a
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
>

<!- additional attributes on form element ->
```

```
<!ATTLIST form
    onsubmit    %Script;            #IMPLIED
    onreset     %Script;            #IMPLIED
>

<!- additional attributes on label element ->

<!ATTLIST label
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
>

<!- additional attributes on input element ->

<!ATTLIST input
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
    onselect    %Script;            #IMPLIED
    onchange    %Script;            #IMPLIED
>

<!- additional attributes on select element ->

<!ATTLIST select
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
    onchange    %Script;            #IMPLIED
>

<!- additional attributes on textarea element ->

<!ATTLIST textarea
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
    onselect    %Script;            #IMPLIED
    onchange    %Script;            #IMPLIED
>
```

```
<!- additional attributes on button element ->

<!ATTLIST button
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
>

<!- additional attributes on body element ->

<!ATTLIST body
    onload      %Script;            #IMPLIED
    onunload    %Script;            #IMPLIED
>
<!- additional attributes on area element ->

<!ATTLIST area
    onfocus     %Script;            #IMPLIED
    onblur      %Script;            #IMPLIED
>

<!ENTITY % XHTML1-frames.module "IGNORE" >
<![%XHTML1-frames.module;[
<!- additional attributes on frameset element ->

<!ATTLIST frameset
    onload      %Script;            #IMPLIED
    onunload    %Script;            #IMPLIED
>
]]>
<!- end of XHTML1-events.mod ->
```

تعطي قيم السمات المختلفة على أساس أنها Script% وتعتبر Script% معامل مرجع كينونة وقد تم تعريفها في XHTML1-names.mod كقيمة تساوي CDATA.

لم يتم تعريف أي من هذه العناصر بعد ويتم التعريف في وحدات نمطية سيتم استيرادها فيما بعد.

الوحدات النمطية للسمات العامة "المشتركة"

توضيح الوحدة النمطية التالية التي تم استيرادها السمات المشتركة في معظم العناصر مثل class وstyle وtitle ويوجد مجموعتين مختلفتين من تلك العناصر وهما مجموعة تخص DTD الثابتة ومجموعة تخص DTD الانتقالية والتي تقدم سمة align تستورد XHTML1-s.dtd الوحدة النمطية XHTML1-attribs.mod الموضحة في تعليمات برمجة ٢٠-٨ ويرمز الحرف T إلى كونها انتقالية.

تعليمات البرمجة ٢٠-٧: XHTML1-attribs.mod: الوحدة النمطية XHTML للسمات المشتركة الثابتة

```
<!- ................................................... ->
<!- XHTML 1.0 Common Attributes Module  ................. ->
<!- file: XHTML1-attribs.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-attribs.mod 1.14 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"
    SYSTEM "XHTML1-attribs.mod"

    Revisions:
# 1999-02-24  changed PE names for attribute classes to *.attrib;
    ........................................................ ->

<!- ii. Common Attributes

    This modules declares many of the common attributes for
    the Strict DTD.
->
```

```
<!ENTITY % Core.attrib
"id          ID                        #IMPLIED
    class       CDATA                  #IMPLIED
    style       %StyleSheet;             #IMPLIED
    title       %Text;                 #IMPLIED"
>

<!ENTITY % I18n.attrib
   "lang         %LanguageCode;          #IMPLIED
    xml:lang     %LanguageCode;          #IMPLIED
    dir          (ltr|rtl)             #IMPLIED"
>

<!- HTML intrinsic event attributes declared previously ->
<!ENTITY % Events.attrib "" >

<!ENTITY % Common.attrib
    "%Core.attrib;
     %I18n.attrib;
     %Events.attrib;" >

<!ENTITY % Align.attrib "" >

<!ENTITY % XLink.attribs  "INCLUDE" >
<![%XLink.attribs;[
<!- XLink attributes for a simple 'a' style link ->

<!ENTITY % Alink.attrib
    "xml:link      CDATA                #FIXED   'simple'
     role          CDATA                #IMPLIED
     inline        CDATA                #FIXED   'true'
     content-role  CDATA                #IMPLIED
     content-title CDATA                #IMPLIED
     show          CDATA                #FIXED   'replace'
     activate      CDATA                #FIXED   'user'
     behavior      CDATA                #IMPLIED"
>
]]>
```

```
<!ENTITY % Alink.attrib  "" >

<!- end of XHTML1-attribs.mod ->
```

تعليمات البرمجة ٢٠-٨: XHTML1-attribs-t.mod: الوحدة النمطية XHTML للسمات المشتركة الانتقالية

```
<!- ................................................... ->
<!- XHTML 1.0 Transitional Attributes Module .......... ->
<!- file: XHTML1-attribs-t.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-attribs-t.mod 1.14 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

 PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Attributes//EN"
 SYSTEM "XHTML1-attribs-t.mod"

    Revisions:
# 1999-01-24  changed PE names for attribute classes to *.attrib;
    ................................................... ->

<!- ii(t). Common Transitional Attributes

    This modules declares the same set of common attributes as
    the Strict version, but additionally includes ATTLIST
    declarations for the additional attribute specifications
    found in the Transitional DTD.
->

<!ENTITY % Core.attrib
    "id          ID                  #IMPLIED
     class       CDATA                  #IMPLIED
```

```
    style      %StyleSheet;          #IMPLIED
    title      %Text;                #IMPLIED"
>

<!ENTITY % I18n.attrib
   "lang        %LanguageCode;        #IMPLIED
    xml:lang    %LanguageCode;        #IMPLIED
    dir         (ltr|rtl)             #IMPLIED"
>

<!- HTML intrinsic event attributes declared previously ->

<!ENTITY % Common.attrib
   "%Core.attrib;
    %I18n.attrib;
    %Events.attrib;"
>

<!- horizontal text alignment ->
<!ENTITY % Align.attrib
   "align   (left|center|right|justify)   #IMPLIED"
>

<!- horizontal and vertical alignment ->
<!ENTITY % IAlign.attrib
   "align   (top|middle|bottom|left|right) #IMPLIED"
>

<!ENTITY % XLink.attribs  "INCLUDE" >
<![%XLink.attribs;[
<!- XLink attributes for a simple anchor link ->

<!ENTITY % Alink.attrib
   "xml:link    CDATA              #FIXED   'simple'
    role        CDATA              #IMPLIED
    inline      CDATA              #FIXED   'true'
    content-role CDATA             #IMPLIED
    content-title CDATA            #IMPLIED
```

```
    show        CDATA                  #FIXED   'replace'
    activate    CDATA                  #FIXED   'user'
    behavior    CDATA                  #IMPLIED"
>
]]>
<!ENTITY % Alink.attrib  "" >

<!- end of XHTML1-attribs-t.mod ->
```

تتشابه الوحدتين النمطيتين في العديد من الأوجه على الرغم من وجود سمة align في DTD الانتقال فهي تعرف معاملات الكينونات للسمات ومجموعات السمات والتي تطبق على أي عناصر HTML وتستخدم معاملات الكينونات داخل تعريفات ATTLIST في وحدات نمطية أخرى.

لنستطيع فهم هذا الجزء سنفترض أننا نقوم بخدعة أحد مطاعم الأكلات السريعة والعمل من نقطة الهدف رجوعاً إلى نقطة البدء بدلاً من نقطة البداية وتقدماً إلى الهدف ونستخدم كينونة Common.attrib كما يلي:

```
<!ENTITY % Common.attrib
    "%Core.attrib;
     %I18n.attrib;
     %Events.attrib;"
>
```

تجمع تلك الكينونات السمات التي تطبق على أي عنصر وتعمل كأول جزء في معظم تعريفات ATTLIST في الوحدات النمطية الفردية مثل.

```
<!ATTLIST address
    %Common.attrib;
>
```

يكون آخر عنصر في تعريف Common.attrib هو%Events.attrib وهو يعرف كسلسلة فارغة في XHTML1-attribs.mod.

```
<!-- HTML intrinsic event attributes declared previously -->
<!ENTITY % Events.attrib "" >
```

تشير التعليقات إلى أنه يمكن التجاوز عن هذا في قاعدة DTD لإضافة سمات للعناصر الموجودة بصفة عادية وبالتحديد فقد تم التجاوز عنها في تعليمات برمجية ٢٠-٦ كما يلي.

```
<!ENTITY % Events.attrib
   "onclick     %Script;            #IMPLIED
    ondblclick  %Script;            #IMPLIED
```

```
    onmousedown  %Script;          #IMPLIED
    onmouseup    %Script;          #IMPLIED
    onmouseover  %Script;          #IMPLIED
    onmousemove  %Script;          #IMPLIED
    onmouseout   %Script;          #IMPLIED
    onkeypress   %Script;          #IMPLIED
    onkeydown    %Script;          #IMPLIED
    onkeyup      %Script;          #IMPLIED"
>
```

تم تعريف معامل مرجــــع كينونـــة Script% فـــي تعليمـــات برمجـــة ٢٠-٤ XHTML1-names.mod على أنها CDATA ويبدو استبدال النص لتعريف Common.attrib كما يلي:

```
%Core.attrib;
    %I18n.attrib;
    onclick      CDATA          #IMPLIED
    ondblclick   CDATA          #IMPLIED
    onmousedown  CDATA          #IMPLIED
    onmouseup    CDATA          #IMPLIED
    onmouseover  CDATA          #IMPLIED
    onmousemove  CDATA          #IMPLIED
    onmouseout   CDATA          #IMPLIED
    onkeypress   CDATA          #IMPLIED
    onkeydown    CDATA          #IMPLIED
    onkeyup      CDATA          #IMPLIED
```

العنصر الثاني إلى الأخير تعريف Common.attrib هو I18n.attrib% يتم تعريفها فـــي نفس الوحدة النمطية مع هذا التعريف.

```
<!ENTITY % I18n.attrib
   "lang        %LanguageCode;        #IMPLIED
    xml:lang    %LanguageCode;        #IMPLIED
    dir         (ltr|rtl)             #IMPLIED"
>
```

تعتبر LanguageCode% معامل مرجـــع كينونـــة وقـــد تـــم تعريفـــه فـــي XHTML1-names.mod كاسم مستعار عن CDATA وبتضمين ذلك تتسع ;Common.attrib% لتصبح.

```
%Core.attrib;
 lang       CDATA; #IMPLIED
 xml:lang   CDATA      #IMPLIED
 dir        (ltr|rtl)  #IMPLIED
 onclick    CDATA      #IMPLIED
```

```
ondblclick  CDATA      #IMPLIED
onmousedown CDATA      #IMPLIED
onmouseup   CDATA      #IMPLIED
onmouseover CDATA      #IMPLIED
onmousemove CDATA      #IMPLIED
onmouseout  CDATA      #IMPLIED
onkeypress  CDATA      #IMPLIED
onkeydown   CDATA      #IMPLIED
onkeyup     CDATA      #IMPLIED
```

آخر معامل مرجع كينونة يتم إمداده هو ;Core.attrib% وقـــد تـــم أيضـــاً تعريفـــه فـــي XHTML1-attribs.mod كما يلي:

```
<!ENTITY % Core.attrib
   "id      ID              #IMPLIED
    class   CDATA           #IMPLIED
    style   %StyleSheet;    #IMPLIED
    title   %Text;          #IMPLIED"
>
```

تتضمن تلك التعريفات معاملين آخرين لمراجع الكينونات وهـــي ;StyleSheet% و;Text% وكل منها يمتك إلى CDATA من التعريفات السابقة في XHTML1-names.mod ولذا يصبـــح الامتداد النهائي ;Common.attrib% هو:

```
id          ID         #IMPLIED
class       CDATA      #IMPLIED
style       CDATA      #IMPLIED
title       CDATA      #IMPLIED
lang        CDATA      #IMPLIED
xml:lang    CDATA      #IMPLIED
dir         (ltr|rtl)  #IMPLIED
onclick     CDATA      #IMPLIED
ondblclick  CDATA      #IMPLIED
onmousedown CDATA      #IMPLIED
onmouseup   CDATA      #IMPLIED
onmouseover CDATA      #IMPLIED
onmousemove CDATA      #IMPLIED
onmouseout  CDATA      #IMPLIED
onkeypress  CDATA      #IMPLIED
onkeydown   CDATA      #IMPLIED
onkeyup     CDATA      #IMPLIED
```



في المثال السابق لم تكن المسافات البيضاء مهمة ولذا فالامتداد الفعلي لكينونة %Common.attrib لم يكن منسق بصورة جيدة ولذا فالمسافات البيضاء غير هامة في التعريفات وكذا فلم يكن ذلك فهما ويجب ضبط المسافات البيضاء يدوياً لوضح الأعمدة أو إدراج فاصل الأسطر عند توسيع معامل مرجع كينونة بطريقة يدوية لرؤية المكتوب به.

لذلك صنف كينونة %Common.attrib; معظم المواد الأخرى في هذا الجزء فلن يظهر %Core.attrib أو%I18N.attrib أو;%Events.attrib مكرراً في وحدات نمطية تالية فهي تماماً مثل الأساليب الخاصة في C++ والتي يمكن تحويلها ولكن ليس فقط من اجل تحسين الأداء.

لا يتم تصنيف سمات ارتباط X في %Common.attrib وهذا لأنة على الرغم من أن العديد من العناصر قد تمتلك سمات ارتباط فإن العديد من العناصر أيضا قد لا تمتك سمات أرتباط ولذا فعند إضافة سمات ارتباط X لعنصر يجب استخدام معامل مرجع كينونة منفصل مثل %Alink.attrib.

الوحدة النمطية لنوع المستند

تستورد XHTML DTD التالية وحدة نمطية تعريف كينونات لعناصر النص التالية مثل p وdiv وblockquote وتلك هي العناصر التي تكون بنية الشجرة الأساسية لمستند HTML صحيح التكوين ويتم تقديم وحدتين نمطيتين آخرين هما وحدة نمطية لمجموعة DTD الثابتة وهي موضحة في تعليمات برمجة ٢٠-٩ XHTML1-model.mod ووحدة نمطية لمجموعة DTD الانتقالية وهي موضحة في تعليمات برمجة ٢٠-١٠ XHTML1-model-t.mod.

تعليمات البرمجة ٢٠-٩: XHTML1-model.mod الوحدة النمطية لنوع المستند الثابت

```
<!- .................................................. ->
<!- XHTML 1.0 Document Model Module .................. ->
<!- file: XHTML1-model.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
```

```
    Reserved. Revision: @(#)XHTML1-model.mod 1.12 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"
    SYSTEM "XHTML1-model.mod"

    Revisions:
    (none)
    ................................................ ->

<!- iii. Document Model

    This modules declares entities describing all text flow
    elements, excluding Transitional elements. This module
    describes the groupings of elements that make up HTML's
    document model.

    HTML has two basic content models:

       %Inline.mix;  character-level elements
       %Block.mix;   block-like elements, eg., paragraphs and lists

    The reserved word '#PCDATA' (indicating a text string) is
    now included explicitly with each element declaration, as
    XML requires that the reserved word occur first in a
    content model specification..
->

<!- ................ Miscellaneous Elements ......... ->

<!- These elements are neither block nor inline, and can
    essentially be used anywhere in the document body ->

<!ENTITY % Misc.class
     "ins | del | script | noscript" >
```

```
<!- ..................... Inline Elements ............. ->

<!ENTITY % Inlstruct.class
      "bdo | br | span" >

<!ENTITY % Inlpres.class  "tt | i | b | big | small | sub | sup" >

<!ENTITY % Inlphras.class
    "em | strong | dfn | code | samp | kbd | var | cite | abbr | acronym |
q" >

<!ENTITY % Inlspecial.class  "a | img | object | map" >

<!ENTITY % Formctrl.class  "input | select | textarea | label | button" >

<!- %Inline.class; includes all inline elements, used as a component in
mixes ->

<!ENTITY % Inline.class
    "%Inlstruct.class;
     | %Inlpres.class;
     | %Inlphras.class;
     | %Inlspecial.class;
     | %Formctrl.class;"
>

<!- %Inline.mix; includes all inline elements, including %Misc.class; ->

<!ENTITY % Inline.mix
    "%Inline.class;
     | %Misc.class;"
>

<!- %Inline-noa.class; includes all non-anchor inlines,
    used as a component in mixes ->

<!ENTITY % Inline-noa.class
    "%Inlstruct.class;
     | %Inlpres.class;
```

```
    | %Inlphras.class;
    | img | object | map
    | %Formctrl.class;"
>

<!- %Inline-noa.mix; includes all non-anchor inlines ->

<!ENTITY % Inline-noa.mix
   "%Inline-noa.class;
    | %Misc.class;"
>

<!- .................... Block Elements .......... ->

<!- In the HTML 4.0 DTD, heading and list elements were
    included in the %block; parameter entity. The
    %Heading.class; and %List.class; parameter entities must
    now be included explicitly on element declarations where
    desired.
->

<!- There are six levels of headings from H1 (the most
     important) to H6 (the least important).
 ->
<!ENTITY % Heading.class "h1 | h2 | h3 | h4 | h5 | h6" >

<!ENTITY % List.class "ul | ol | dl" >

<!ENTITY % Blkstruct.class "p | div" >

<!ENTITY % Blkpres.class "hr" >

<!ENTITY % Blkphras.class "pre | blockquote | address" >

<!ENTITY % Blkform.class "form | fieldset" >

<!ENTITY % Blkspecial.class "table" >
```

```
<!- %Block.class; includes all block elements,
    used as an component in mixes ->

<!ENTITY % Block.class
    "%Blkstruct.class;
     | %Blkpres.class;
     | %Blkphras.class;
     | %Blkform.class;
     | %Blkspecial.class;"
>

<!- %Block.mix; includes all block elements plus %Misc.class;
->

<!ENTITY % Block.mix
    "%Block.class;
     | %Misc.class;"
>

<!- %Block-noform.class; includes all non-form block elements,
    used as a component in mixes ->

<!ENTITY % Block-noform.class
    "%Blkstruct.class;
     | %Blkpres.class;
     | %Blkphras.class;
     | %Blkspecial.class;"
>

<!- %Block-noform.mix; includes all non-form block elements,
    plus %Misc.class; ->

<!ENTITY % Block-noform.mix
    "%Block-noform.class;
     | %Misc.class;"
>

<!- ............... All Content Elements ......... ->
```

```
<!- %Flow.mix; includes all text content, block and inline ->

<!ENTITY % Flow.mix
      "%Heading.class;
     | %List.class;
     | %Block.class;
     | %Inline.class;
     | %Misc.class;"
>

<!- end of XHTML1-model.mod ->
```

تعليمات البرمجة ٢٠-١٠: XHTML1-model-t.mod الوحدة النمطية لنموذج المستند الانتقالي

```
<!- .................................................. ->
<!- XHTML 1.0 Transitional Text Markup Module .......... ->
<!- file: XHTML1-model-t.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-model-t.mod 1.14 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Document
    Model//EN" SYSTEM "XHTML1-model-t.mod"

    Revisions:
#1999-01-14  rearranged forms and frames PEs, adding
%Blkform.class;
    .................................................. ->

<!- iii(t). Transitional Document Model
```

```
    This modules declares entities describing all text flow
    elements, including Transitional elements. This module
    describes the groupings of elements that make up HTML's
    document model.

    HTML has two basic content models:

       %Inline.mix;  character-level elements
       %Block.mix;   block-like elements, eg., paragraphs and lists

    The reserved word '#PCDATA' (indicating a text string) is
    now included explicitly with each element declaration, as
    XML requires that the reserved word occur first in a
content model specification..
->

<!- ................ Miscellaneous Elements ............... ->

<!- These elements are neither block nor inline, and can
    essentially be used anywhere in the document body ->

<!ENTITY % Misc.class
      "ins | del | script | noscript" >

<!- ................... Inline Elements ............. ->

<!ENTITY % Inlstruct.class
      "bdo | br | span" >

<!ENTITY % Inlpres.class
"tt | i | b | u | s | strike | big | small | font | basefont
   | sub | sup" >

<!ENTITY % Inlphras.class
      "em | strong | dfn | code | samp | kbd | var | cite
      | abbr | acronym | q" >
```

```
<![%XHTML1-frames.module;[
<!- %Inlspecial.class; includes iframe in Frameset DTD version
->

<!ENTITY % Inlspecial.class  "a | img | applet | object | map
                         | iframe">
]]>

<!ENTITY % Inlspecial.class  "a | img | applet | object | map">

<!ENTITY % Formctrl.class  "input | select | textarea | label
                          | button">

<!- %Inline.class; includes all inline elements, used
                as a component in mixes ->

<!ENTITY % Inline.class
    "%Inlstruct.class;
     | %Inlpres.class;
     | %Inlphras.class;
     | %Inlspecial.class;
     | %Formctrl.class;"
>

<!- %Inline.mix; includes all inline elements,
              including %Misc.class; ->

<!ENTITY % Inline.mix
    "%Inline.class;
     | %Misc.class;"
>

<!- %Inline-noa.class; includes all non-anchor inlines,
    used as a component in mixes ->

<!ENTITY % Inline-noa.class
    "%Inlstruct.class;
     | %Inlpres.class;
```

```
        | %Inlphras.class;
        | img | applet | object | map
        | %Formctrl.class;"
>

<!- %Inline-noa.mix; includes all non-anchor inlines ->

<!ENTITY % Inline-noa.mix
      "%Inline-noa.class;
       | %Misc.class;"
>

<!- .................... Block Elements .......... ->

<!- In the HTML 4.0 DTD, heading and list elements were
     included in the %block; parameter entity. The
     %Heading.class; and %List.class; parameter entities must
     now be included explicitly on element declarations where
     desired.
->

<!-  There are six levels of headings from h1 (the most important)
      to h6 (the least important).
->
<!ENTITY % Heading.class  "h1 | h2 | h3 | h4 | h5 | h6">

<!ENTITY % List.class  "ul | ol | dl | menu | dir" >

<!ENTITY % Blkstruct.class  "p | div" >

<!ENTITY % Blkpres.class  "center | hr" >

<!ENTITY % Blkphras.class  "pre | blockquote | address" >

<!ENTITY % Blkform.class  "form | fieldset" >

<![%XHTML1-frames.module;[
<!- Blkspecial.class includes noframes in Frameset DTD version ->
```

```
<!ENTITY % Blkspecial.class  "noframes | table" >
]]>

<!ENTITY % Blkspecial.class  "table" >

<!- %Block.class; includes all block elements,
   used as an component in mixes ->

<!ENTITY % Block.class
   "%Blkstruct.class;
    | %Blkpres.class;
    | %Blkphras.class;
    | %Blkform.class;
    | %Blkspecial.class;"
>

<!- %Block.mix; includes all block elements plus %Misc.class; ->

<!ENTITY % Block.mix
   "%Block.class;
    | %Misc.class;"
>

<!- %Block-noform.class; includes all non-form block elements,
   used as a component in mixes ->

<!ENTITY % Block-noform.class
   "%Blkstruct.class;
    | %Blkpres.class;
    | %Blkphras.class;
    | %Blkspecial.class;"
>

<!- %Block-noform.mix; includes all non-form block elements,
   plus %Misc.class; ->

<!ENTITY % Block-noform.mix
```

```
    "%Block-noform.class;
     | %Misc.class;"
>

<!- ............... All Content Elements ........... ->

<!- %Flow.mix; includes all text content, block and inline ->

<!ENTITY % Flow.mix
      "%Heading.class;
     | %List.class;
     | %Block.class;
     | %Inline.class;
     | %Misc.class;"
>
<!- end of XHTML1-model-t.mod ->
```

لا تعتبر العناصر هي نفسها التي تم تعريفها في تلك الوحدتين النمطيتين ولكنها كينونات تستخدم في نماذج المحتوي لهذه العناصر والعناصر التي تحتويهم ويأتي التعرف الفعلي للعناصر فيما بعد.

تقسم الوحدات النمطية إلى مقاطع منطقية يشار إليها باستخدام التعليقات أول جزء هو Miscellaneous Elements "العناصر المتعددة" وهو يعرف معامل كينونة misc.class للاربعة عناصر التي قد تظهر إما كعناصر محولة أو عناصر كتل.

```
<!ENTITY % Misc.class
      "ins | del | script | noscript" >
```

يعرف جزء العناصر المحولة العناصر المحولة في HTML وهم الذي قد يحتوي على عناصر مستوى الكتلة ويختلف DTD الثابتة والانتقالية في تحديد العناصر التي يتم تضمينها ومع ذلك فكلاهما يقسما العناصر المحولة إلى فئات هيكلية (Inlstruct.class) وفئات تقدمية (Inlphras.class) وفئات النوذج (Formctrl.class) تدمج معاملات الكينونات المتوسطة التكوين معامل كينونة Inline.class والتي تدرج كل العناصر التي قد تظهر كعناصر محولة ثم يتم دمج Inline.class% مع معامل مرجع الكينونة المعرف سابقا Misc.class% لإنشاء معامل كينونة Inline.mix الذي يتضمن كلا من العناصر المتعددة والمحولة.

```
<!ENTITY % Inline.mix
    "%Inline.class;
     | %Misc.class;"
>
```

يعرف أيضا معامل كينونة مماثل يسمى Inline-noa.class وترمز noa إلى ليس عنصر ويترك هذا العنصر لأنه سيكون مطلوباً في مكان آخر عندما يتم تعريف كينونات مستوى الكتلة ويؤدي تضمينه هنا إلى نماذج محتوى غامضة وهو ليس مشكلة ولكن يفضل تجنبها كلما أمكن.

يدرج جزء عناصر الكتلة الأنواع المختلفة لعناصر مستوى الكتلة ويعرف معامل الكينونة لكل منهما وهذا يتصاعد إلى أن يصل إلى معامل مرجع كينونة Block.class% والذي يدرج كل عناصر مستوى الكتلة Flow.mix% الذي يدرج كل عناصر الكتلة والعناصر المحولة.

تعريف معاملات لكينونات لرؤوس الصفحة من h1 إلى h6 (Heading.class) والقوائم (List.class) ويتضمن معاملات كينونات مستوي الكتلة كتل البنية p (Blkstruct.class) div والكتل التقدمية وبالتحديد نماذج hr (Blkpres.class) ومجموعات الحقول (Blkform.class) والجداول (Blkspecial.class) ويتم دمج كل ذلك في معامل كينونة Block.class ويدمج هذا مع معامل كينونة Misc. class.لتكوين معامل كينونة Block.mix الذي يحتوي على كلا من العناصر المتعددة وعناصر مستوى الكتلة وأخيراً يتم تعريف كينونات Block-noform.class وBlock-noform.mix لتستخدم عندما تكون كل عناصر مستوى الكتلة ماعدا النماذج مطلوبه.

يعرف الجزء الأخير عناصر المحتوى Flow.mix ويجمع هذا الجزء كل ما سبق وهم عناصر الكتلة والعناصر المحولة وعناصر رأس الصفحة وعناصر القائمة والعناصر المتعددة.

```
<!ENTITY % Flow.mix
     "%Heading.class;
     | %List.class;
     | %Block.class;
     | %Inline.class;
     | %Misc.class;"
>
```

الوحدة النمطية الهيكلية الضمنية

تستخدم الوحدة النمطية التالية XHTML1-inlstruct.mod الموضحة في تعليمات البرمجة ٢٠-١١ بواسطة كلاً من DTD الثابت والانتقالي لتعريف عناصر البنية المحولة br ,bdo وdel وins.

تعليمات البرمجة ٢٠-١١: XHTML1-inlstruct.mod الوحدة النمطية للبنية الضمنية

```
<!- ................................................ ->
<!- XHTML 1.0 Inline Phrasal Module .................. ->
<!- file: XHTML1-inlstruct.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-inlstruct.mod 1.10 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    Identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
    SYSTEM "XHTML1-inlstruct.mod"

    Revisions:
    (none)
    ................................................ ->

<!- c1. Inline Structural

      bdo, br, del, ins, span
->

<!ENTITY % Bdo.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT bdo %Bdo.content; >
<!ATTLIST bdo
    %Core.attrib;
    lang        %LanguageCode;        #IMPLIED
    dir         (ltr|rtl)             #REQUIRED
>

<!ENTITY % Br.content "EMPTY" >
<!ELEMENT br %Br.content; >
```

```
<!ATTLIST br
    %Core.attrib;
>

<!ENTITY % Del.content "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT del %Del.content; >
<!ATTLIST del
    %Common.attrib;
    cite        %URI;              #IMPLIED
    datetime    %Datetime;         #IMPLIED
>

<!ENTITY % Ins.content "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT ins %Ins.content; >
<!ATTLIST ins
    %Common.attrib;
    cite        %URI;              #IMPLIED
    datetime    %Datetime;         #IMPLIED
>

<!ENTITY % Span.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT span %Span.content; >
<!ATTLIST span
    %Common.attrib;
>

<!- end of XHTML1-inlstruct.mod ->
```

تبدأ تلك الوحدة النمطية في استخدام معاملات الكينونات التي قامت الوحدات النمطية المتعددة الأخيرة بتعريفها وبالتحديد فهي تعرف سمات del وins وspan على أساس أنها %Common.attrib وتعرف سمات bdo وbr على أساس أنها %Core.attrib وتستخدم أيضا العديد من الأسماء المستعارة CDATA من XHTML1-names.mod وبالتحديد %LanguageCode;, و%URI; و%Datetime;.

لاحظ أن يتم إعطاء نماذج المحتوى للعناصر ككينونات معرفة محليا ومثال على ذلك:

```
<!ENTITY % Span.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT span %Span.content; >
```

لماذا لا يتم تعريفهم دون معامل مرجع كينونة مثل ما يلي:

```
<!ELEMENT span ( #PCDATA | %Inline.mix; )* >
```

السبب في ذلك هو أن استخدام معامل مرجع الكينونة يتيح لوحدات نمطية أخرى تجاوز نموذج المحتوى وعموماً فليس من الضروري أن تكون تلك هي الوحدات النمطية المستخدمة في هذا المثال ولكن قد تكون الوحدات النمطية من تطبيقات XML مختلفة تماماً والتي قد يتم دمجها مع وحدات XHTML النمطية.

الوحدة النمطية التقدمية المحولة

تستخدم الوحدة النمطية التالية XHTML1-inlpres.mod الموضحة في تعليمات برمجة ٢٠-١٢ من قبل كلا من DTD الثابت الانتقالي لتعريف العناصر التقدمية b وbig وi, وsmall وsub وsup وtt.

تعليمات البرمجة ٢٠-١٢: XHTML1-inlpres.mod الوحدة النمطية التقدمية الضمنية

```
<!- ................................................... ->
<!- XHTML 1.0 Inline Presentational Module ............. ->
<!- file: XHTML1-inlpres.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-inlpres.mod 1.13 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline
    Presentational//EN" SYSTEM "XHTML1-inlpres.mod"

    Revisions:
    (none)
    ................................................... ->
```

```
<!- c3. Inline Presentational

      b, big, i, small, sub, sup, tt

   A conditional section includes additional declarations for
   the Transitional DTD

      basefont, font, s, strike, u
->

<!ENTITY % B.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT b  %B.content; >
<!ATTLIST b
    %Common.attrib;
>

<!ENTITY % Big.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT big  %Big.content; >
<!ATTLIST big
    %Common.attrib;
>

<!ENTITY % I.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT i  %I.content; >
<!ATTLIST i
    %Common.attrib;
>

<!ENTITY % Small.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT small  %Small.content; >
<!ATTLIST small
    %Common.attrib;
>

<!ENTITY % Sub.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT sub  %Sub.content; >
<!ATTLIST sub
```

```
    %Common.attrib;
>

<!ENTITY % Sup.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT sup  %Sup.content; >
<!ATTLIST sup
    %Common.attrib;
>

<!ENTITY % Tt.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT tt    %Tt.content; >
<!ATTLIST tt
    %Common.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % Basefont.content  "EMPTY" >
<!ELEMENT basefont  %Basefont.content; >
<!ATTLIST basefont
    id         ID                  #IMPLIED
    size       CDATA               #REQUIRED
    color      %Color;             #IMPLIED
    face       CDATA               #IMPLIED
>

<!ENTITY % Font.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT font  %Font.content; >
<!ATTLIST font
    %Core.attrib;
    %I18n.attrib;
    size       CDATA               #IMPLIED
    color      %Color;             #IMPLIED
    face       CDATA               #IMPLIED
>

<!ENTITY % S.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT s  %S.content; >
<!ATTLIST s
```

```
    %Common.attrib;
>

<!ENTITY % Strike.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT strike  %Strike.content; >
<!ATTLIST strike
    %Common.attrib;
>

<!ENTITY % U.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT u  %U.content; >
<!ATTLIST u
    %Common.attrib;
>

]]>

<!- end of XHTML1-inlpres.mod ->
```

تنتج إحدى الطرق في هذا الملف تعريف عناصر basefont و font و s و strike و u لتعمل بها DTD الانتقالية وليس DTD الثابتة وقد تم تجميع تعريفات تلك العناصر وسماتها في التكوين التالي.

```
<![%XHTML.Transitional;[
  <!- basefont, font, s, strike, and u declarations ->
]]>
```

تذكر أن XHTML-t.dtd قد عرفت معامل الكينونة XHTML.Transitional على أنها INCLUDE ولكن XHTML-s.dtd عرفت على أنها INCLUDE ولهذا يتم تضمين تلك التعريفات بواسطة DTD الانتقالي ويتم تجاهلها بواسطة DTD الثابت.

الوحدة النمطية للعبارات الضمنية

تستخدم الوحدة النمطية التالية XHTML1-inlphras.mod, الموضحة في تعليمات برمجة ٢٠-١٣ بواسطة كلا من DTD الثابت والانتقالي لتعريف عناصر العبارات الضمنية وهي abbr و acronym و cite و code و em و dfn, و kbd, و q و samp و strong و and و var.

تعليمات البرمجة ٢٠-١٣ : XHTML1-inlphras.mod الوحدة النمطية العبارات الضمنية

```
<!- ................................................... ->
<!- XHTML 1.0 Inline Phrasal Module ..................... ->
<!- file: XHTML1-inlphras.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-inlphras.mod 1.14 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
    SYSTEM "XHTML1-inlphras.mod"

    Revisions:
#1999-01-29  moved bdo, br, del, ins, span to inline
           structural module
    ................................................. ->

<!- c2. Inline Phrasal

      abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
->

<!ENTITY % Abbr.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT abbr  %Abbr.content; >
<!ATTLIST abbr
     %Common.attrib;
>

<!ENTITY % Acronym.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT acronym  %Acronym.content; >
<!ATTLIST acronym
```

```
    %Common.attrib;
>

<!ENTITY % Cite.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT cite  %Cite.content; >
<!ATTLIST cite
    %Common.attrib;
>

<!ENTITY % Code.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT code  %Code.content; >
<!ATTLIST code
    %Common.attrib;
>

<!ENTITY % Dfn.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT dfn  %Dfn.content; >
<!ATTLIST dfn
    %Common.attrib;
>

<!ENTITY % Em.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT em  %Em.content; >
<!ATTLIST em
    %Common.attrib;
>

<!ENTITY % Kbd.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT kbd  %Kbd.content; >
<!ATTLIST kbd
    %Common.attrib;
>

<!ENTITY % Q.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT q  %Q.content; >
<!ATTLIST q
    %Common.attrib;
    cite        %URI;                  #IMPLIED
```

```
>

<!ENTITY % Samp.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT samp %Samp.content; >
<!ATTLIST samp
    %Common.attrib;
>

<!ENTITY % Strong.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT strong %Strong.content; >
<!ATTLIST strong
    %Common.attrib;
>

<!ENTITY % Var.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT var %Var.content; >
<!ATTLIST var
    %Common.attrib;
>

<!- end of XHTML1-inlphras.mod ->
```

باستثناء q فإن لكل العناصر المحولة في تلك الوحدة النمطية نماذج محتويات مشابهة وقوائم سمات متشابهة وقد يحتوي كلها على سمات #PCDATA | %Inline.mix و%Common.attrib وقد يكون لعنصر q كل تلك السمات أيضا وقد يكون له سمة اختيارية إضافية أخرى هي cite وتحتوي على URI يشير إلى مصدر quotation.

يوضح هذا المثال قوة اتجاه كينونة المعامل بصورة جيدة وبدون مراجع كينونات المعاملات تظهر هذه الوحدة النمطية أطول وأصعب في الفهم.

الوحدة النمطية لبنية الكتلة

تستخدم الوحدة النمطية التالية وهي XHTML1-blkstruct.mod, الموضحة في تعليمات برمجة ٢٠-١٤ بواسطة DTD الثابت والقياسي لتعريف عناصر بنية مستوى الكتلة p وdiv.

تعليمات البرمجة ٢٠-١٤: XHTML1-blkstruct.mod الوحدة النمطية للعبارات الضمنية

```
<!– ...............................................................–>
<!– XHTML 1.0 Block Structural Module  ................. –>
<!– file: XHTML1-blkstruct.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-blkstruct.mod 1.10 99/04/01
    SMI

This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
    SYSTEM "XHTML1-blkstruct.mod"

    Revisions:
    (none)
    ................................................... –>

<!– b1. Block Structural

      div, p
–>

<!ENTITY % Div.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT div  %Div.content; >
<!ATTLIST div
     %Common.attrib;
     %Align.attrib;
>

<!ENTITY % P.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT p  %P.content; >
```

```
<!ATTLIST p
    %Common.attrib;
>

<!-- end of XHTML1-blkstruct.mod -->
```

الوحدة النمطية التقدمية للكتلة

تعرف الوحدة النمطية التالية وهي XHTML1-blkpres.mod الموضحة في تعليمـــات برمجـــة ٢١-٢٥ عناصر مستوى كتلة البنية hr وcenter لكل من DTD الثابت والانتقالي.

تعليمات البرمجة ٢٠-١٥: XHTML1-blkpres.mod الوحدة النمطية التقدمية المضمنة

```
<!-- ............................................... -->
<!-- XHTML 1.0 Block Presentational Module .............. -->
<!-- file: XHTML1-blkpres.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-blkpres.mod 1.15 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block
    Presentational//EN" SYSTEM "XHTML1-blkpres.mod"

    Revisions:
# 1999-01-31  added I18n.attrib to hr (errata)
    ............................................... -->

<!-- b3. Block Presentational

     hr
```

```
    A conditional section includes additional declarations
    for the Transitional DTD

      center
->

<!ENTITY % Hr.content  "EMPTY" >
<!ELEMENT hr  %Hr.content; >
<!ATTLIST hr
    %Core.attrib;
    %I18n.attrib;
    %Events.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % Center.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT center  %Center.content; >
<!ATTLIST center
    %Common.attrib;
>

<!- additional attributes on hr ->
<!ATTLIST hr
    align       (left|center|right)      #IMPLIED
    noshade       (noshade)               #IMPLIED
    size        %Pixels;                #IMPLIED
    width        %Length;                 #IMPLIED
>
]]>

<!- end of XHTML1-blkpres.mod ->
```

يصغر عنصر center في HTML 4.0 ولذا فقد تم وضعة فـــي قطـــاع <![%XHTML. Transitional;[التي ستضمن بواسطة DTD الانتقالي ويتجاهلها DTD الثابت ويتضمن كـــلا منها عنصر hr ومع ذلك فبعض وليس كل السمات يتم تصغيرها في وبالتـــالي فلـــها تعريفيـــن ATTLIST واحدة للسمة غير المصغرة وأخرى للسمة المصغرة ويوضـــع تعريـــف ATTLIST للسمة المصغرة في قطاع <![%XHTML.Transitional;[ ]]> ولذا يتم تجاهلها من قبل DTD الثابت.

الوحدة النمطية لكتلة العبارات

تستخدم الوحدة النمطية التالية والموضحة في تعليمات برمجة ٢٠-١٦ وهي -XHTML1 blkphras.mod بواسطة كلا من DTD الانتقالي والثابت لتعريف عناصر عبارات مستوى الكتلة وهم blockquote, pre, h1, h2, h3, h4, h5.

تعليمات البرمجة ٢٠-١٦: XHTML1-blkphras.mod الوحدة النمطية لكتلة العبارات

```
<!- ................................................... ->
<!- XHTML 1.0 Block Phrasal Module ................... ->
<!- file: XHTML1-blkphras.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved. Revision: @(#)XHTML1-blkphras.mod 1.13 99/04/01
    SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
    SYSTEM "XHTML1-blkphras.mod"

    Revisions:
# 1998-11-10  removed pre exclusions - content model
            changed to mimic HTML 4.0
# 1999-01-29  moved div and p to block structural module
    ................................................... ->

<!- b2. Block Phrasal

      address, blockquote, pre, h1, h2, h3, h4, h5, h6
->

<!ENTITY % Address.content  "( #PCDATA | %Inline.mix; )*" >
```

```
<!ELEMENT address  %Address.content; >
<!ATTLIST address
    %Common.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % Blockquote.content  "( %Flow.mix; )*" >
]]>
<!ENTITY % Blockquote.content
   "( %Heading.class;
    | %List.class;
    | %Block.mix; )+"
>

<!ELEMENT blockquote  %Blockquote.content; >
<!ATTLIST blockquote
    %Common.attrib;
    cite        %URI;                 #IMPLIED
>

<!ENTITY % Pre.content
   "( #PCDATA | tt | i | b
    | %Inlstruct.class; | %Inlphras.class;
    | a | script | map
    | %Formctrl.class; )*"
>

<!ELEMENT pre  %Pre.content; >
<!ATTLIST pre
    %Common.attrib;
    xml:space   CDATA                 #FIXED "preserve"
>

>

<!ENTITY % Textarea.content  "( #PCDATA )" >
<!ELEMENT textarea  %Textarea.content; >
<!ATTLIST textarea
```

```
    %Common.attrib;
    name        CDATA                 #IMPLIED
    rows        %Number;              #REQUIRED
    cols        %Number;              #REQUIRED
    disabled    (disabled)            #IMPLIED
    readonly    (readonly)            #IMPLIED
    tabindex    %Number;              #IMPLIED
    accesskey   %Character;           #IMPLIED
>

<!- #PCDATA is to solve the mixed content problem, per
   specification only whitespace is allowed there!
->

<!ENTITY % Fieldset.content
  "( #PCDATA | legend | %Flow.mix; )*" >
<!ELEMENT fieldset  %Fieldset.content; >
<!ATTLIST fieldset
    %Common.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % LegendAlign.attrib
   "align      (top|bottom|left|right)  #IMPLIED" >
]]>
<!ENTITY % LegendAlign.attrib  "" >

<!ENTITY % Legend.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT legend  %Legend.content; >
<!ATTLIST legend
    %Common.attrib;
    accesskey   %Character;           #IMPLIED
    %LegendAlign.attrib;
>

<!ENTITY % Button.content
   "( #PCDATA
   | %Heading.class;
```

```
    | %List.class;
    | %Inlpres.class;
    | %Inlphras.class;
    | %Block-noform.mix;
    | img | object | map )*"
>
<!ELEMENT button  %Button.content; >
<!ATTLIST button
    %Common.attrib;
    name        CDATA                  #IMPLIED
    value       CDATA                  #IMPLIED
    type        (button|submit|reset)  'submit'
    disabled    (disabled)             #IMPLIED
    tabindex    %Number;               #IMPLIED
    accesskey   %Character;            #IMPLIED
>
<!- end of forms.mod ->
The Table Module<!- .................. Heading Elements .............. ->

<!ENTITY % Heading.content  "( #PCDATA | %Inline.mix; )*" >

<!ELEMENT h1  %Heading.content; >
<!ATTLIST h1
    %Common.attrib;
    %Align.attrib;
>

<!ELEMENT h2  %Heading.content; >
<!ATTLIST h2
    %Common.attrib;
    %Align.attrib;
>

<!ELEMENT h3  %Heading.content; >
<!ATTLIST h3
    %Common.attrib;
    %Align.attrib;
>
```

```
<!ELEMENT h4  %Heading.content; >
<!ATTLIST h4
%Common.attrib;
    %Align.attrib;
>

<!ELEMENT h5  %Heading.content; >
<!ATTLIST h5
    %Common.attrib;
    %Align.attrib;
>

<!ELEMENT h6  %Heading.content; >
<!ATTLIST h6
    %Common.attrib;
    %Align.attrib;
>

<!- end of XHTML1-blkphras.mod ->
```

يفصل قطاع <![%XHTML.Transitional;[ ]]> تعريفات كلا مـــن DTD الثـــابت عـــن الانتقالي وما يلي هو نموذج محتوى عنصر blockquote الذي يتم ضبطه اعتمـــاداً علـــى أي DTD يتم استخدامه في السطور التالية:

```
<![%XHTML.Transitional;[
<!ENTITY % Blockquote.content  "( %Flow.mix; )*" >
]]>
<!ENTITY % Blockquote.content
    "( %Heading.class;
     | %List.class;
     | %Block.mix; )+"
```

يستخدم أول تعريف Blockquote.content مع DTD الانتقالي فقط فإذا تم تضمينها فإنـــها تأخذ الأسبقية على إعادة التعريف الثانية ومع DTD الثابت فإن التعريف الثاني هو فقـــط الـــذي يظهر أو يستخدم.

الوحدة النمطية للغة الحوار

تستخدم الوحدة النمطية التالية الموضحة في تعليمات برمجـــة ٢٠-١٧ XHTML1-script.mod بواسطة كل من DTD الثابت والافتراضي لتعريف عناصر script وnoscript.

تعليمات البرمجة ٢٠-١٧: XHTML1-script.mod الوحدة النمطية للغة الحوار

```
<!- .................................................. ->
<!- XHTML 1.0 Document Scripting Module ............... ->
<!- file: XHTML1-script.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-script.mod 1.13 99/04/01 SMI

    This DTD module is identified by the PUBLIC
    and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
    SYSTEM "XHTML1-script.mod"

    Revisions:
# 1999-01-30  added xml:space to script
# 1999-02-01  removed for and event attributes from script
    ..................................................... ->

<!- d4. Scripting

      script, noscript
->

<!ENTITY % Script.content  "( #PCDATA )" >
<!ELEMENT script  %Script.content; >
<!ATTLIST script
    charset     %Charset;             #IMPLIED
    type        %ContentType;         #REQUIRED
```

```
    src         %URI;                 #IMPLIED
    defer        (defer)              #IMPLIED
    xml:space    CDATA                  #FIXED 'preserve'
>

<!ENTITY % Noscript.content
    "( %Heading.class;
     | %List.class;
     | %Block.mix; )+"
>
<!ELEMENT noscript  %Noscript.content; >
<!ATTLIST noscript
    %Common.attrib;
>

<!- end of XHTML1-script.mod ->
The Stylesheets Module
```

الوحدة النمطية لأوراق النمط

تستخدم الوحدة النمطية التالية الموضحة في تعليمات برمجـــة ٢٠-١٨ XHTML1-style.mod بواسطة DTD الثابت والانتقالي لتعريف عنصر واحد هو style.

تعليمات البرمجة ٢٠-١٨: XHTML1-style.mod الوحدة النمطية لأوراق النمط

```
<!- ................................................... ->
<!- XHTML 1.0 Document Stylesheet Module  ................ ->
<!- file: XHTML1-style.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-style.mod 1.13 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:
```

```
    PUBLIC "-//W3C//DTD XHTML 1.0 Stylesheets//EN"
    SYSTEM "XHTML1-style.mod"

    Revisions:
# 1999-01-30  added xml:space to style
    ........................................................ ->

<!- d5. Stylesheets

      style
->

<!ENTITY % Style.content  "( #PCDATA )" >
<!ELEMENT style  %Style.content; >
<!ATTLIST style
     %I18n.attrib;
     type        %ContentType;          #REQUIRED
     media       %MediaDesc;            #IMPLIED
     title       %Text;                 #IMPLIED
     xml:space   CDATA                  #FIXED 'preserve'
>

<!- end of XHTML1-style.mod ->
```

The Image Module

الوحدة النمطية للصور

تستخدم الوحدة النمطية الموضحة في تعليمات برمجة ٢٠-١٩ XHTML1- بواسطة DTD الثابت والانتقالي لتعريف عنصر واحد هو Ing.

تعليمات البرمجة ٢٠-١٩ : XHTML1-image.mod وحدة الصور النمطية

```
<!- ................................................... ->
<!- XHTML 1.0 Images Module  ............................ ->
<!- file: XHTML1-image.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
```



```
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-image.mod 1.15 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
    SYSTEM "XHTML1-image.mod"

    Revisions:
# 1999-01-31  corrected transitional attributes (errata)
    .................................................... ->

<!- d3.1. Images

      img
->

<!- To avoid problems with text-only UAs as well as
    to make image content understandable and navigable
    to users of non-visual UAs, you need to provide
    a description with ALT, and avoid server-side image maps
->

<!ENTITY % Img.content  "EMPTY" >
<!ELEMENT img  %Img.content; >
<!ATTLIST img
    %Common.attrib;
    src          %URI;          #REQUIRED
    alt          %Text;         #REQUIRED
    longdesc     %URI;          #IMPLIED
    height       %Length;       #IMPLIED
    width        %Length;       #IMPLIED
    usemap       %URI;          #IMPLIED
    ismap        (ismap)        #IMPLIED
>
```

```
<!- USEMAP points to a MAP element which may be in this
    document or an external document, although the latter is
    not widely supported
->

<![%XHTML.Transitional;[
<!- additional Transitional attributes ->
<!ATTLIST img
     %IAlign.attrib;
     border      %Pixels;          #IMPLIED
     hspace      %Pixels;          #IMPLIED
     vspace      %Pixels;          #IMPLIED
>
]]>

<!- end of XHTML1-image.mod ->
```

لاحظ أن سمة alt تطلب على img وينتج عن حذفها خطأ في صحة التعليمات.

وحدة الأطر النمطية

يستورد كلا من DTD الثابت والانتقالي وحـدات الأطـر النمطيـة XHTML1-frames.mod الموضحة في تعليمات برمجة ٢٠-٢٠ وتعرف الوحدة النمطية العناصر والسـمات المسـتخدمة على صفحات الويب باستخدام الأطر وبالتحديد تقـوم بتعريـف عنـاصر frameset وframe وnoframes وiframe وقوائم السمات المصاحب لها.

ما يلي يوجز هذا الاستيراد:

```
<![%XHTML1-frames.module;[
  <!" frames declarations ">
]]>
```

بالتالي لا تحدث تلك الاستيرادات إلا إذا تم تقديـم معـامل مرجـع كينونـة %XHTML1-frames.module إلى INCLUDE والذي لا يحدث إلا في حالة استخدام مجموعة أطر DTD.

تعليمات البرمجة ٢٠-٢٠: XHTML1-image.mod وحدة الأطر النمطية

```
<!- ............................................... ->
<!- XHTML 1.0 Frames Module ....................... ->
<!- file: XHTML1-frames.mod
```

```
This is XHTML 1.0, an XML reformulation of HTML 4.0.
Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
Reserved.
Revision: @(#)XHTML1-frames.mod 1.15 99/04/01 SMI

This DTD module is identified by the PUBLIC and SYSTEM
identifiers:

PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
SYSTEM "XHTML1-frames.mod"

Revisions:
#1999-01-14  transferred 'target' attribute on 'a' from linking module
..................................................... ->

<!- a2. Frames

      frame, frameset, iframe, noframes
->

<!- The content model for HTML documents depends on whether
    the HEAD is followed by a FRAMESET or BODY element. The
    widespread omission of the BODY start tag makes it
    impractical to define the content model without the use of
    a conditional section.
->

<!ENTITY % Frameset.content  "(( frameset | frame )+, noframes? )" >
<!ELEMENT frameset  %Frameset.content; >
<!ATTLIST frameset
    %Core.attrib;
    rows        %MultiLengths;        #IMPLIED
    cols        %MultiLengths;        #IMPLIED
>

<!- reserved frame names start with "_" otherwise starts with
letter ->
```

```
<!ENTITY % Frame.content  "EMPTY" >
<!ELEMENT frame %Frame.content; >
<!ATTLIST frame
    %Core.attrib;
    longdesc     %URI;         #IMPLIED
    name         CDATA         #IMPLIED
    src          %URI;         #IMPLIED
    frameborder  (1|0)         '1'
    marginwidth  %Pixels;      #IMPLIED
    marginheight %Pixels;      #IMPLIED
    noresize     (noresize)    #IMPLIED
    scrolling    (yes|no|auto) 'auto'
>

<!- Inline Frames ......................... ->

<!ENTITY % Iframe.content  "( %Flow.mix; )*" >
<!ELEMENT iframe  %Iframe.content; >
<!ATTLIST iframe
    %Core.attrib;
    longdesc     %URI;                #IMPLIED
    name         CDATA                #IMPLIED
    src          %URI;                #IMPLIED
    frameborder  (1|0)                '1'
    marginwidth  %Pixels;             #IMPLIED
    marginheight %Pixels;             #IMPLIED
    scrolling    (yes|no|auto)        'auto'
    %IAlign.attrib;
    height       %Length;             #IMPLIED
    width        %Length;             #IMPLIED
>

<!- changes to other declarations ................... ->

<!- redefine content model for html element,
   substituting frameset for body ->
<!ENTITY % Html.content "( head, frameset )" >
```

```
<!- alternate content container for non frame-based rendering
->

<!ENTITY % Noframes.content "( body )">
<!- in HTML 4.0 was "( body ) -( noframes )"
    exclusion on body ->
<!ELEMENT noframes  %Noframes.content; >
<!ATTLIST noframes
     %Common.attrib;
>

<!- add 'target' attribute to 'a' element ->
<!ATTLIST a
     target      %FrameTarget;          #IMPLIED
>

<!- end of XHTML1-frames.mod ->
```

لا يوجد ما يمكن قوله بشأن تلك التعريفات حيث لا يتطلب إضافة أطر إلـــى DTD التجـــاوز عن معاملات كينونات سابقة ومن الأشياء غير الاعتبارية في تلك الوحدة النمطية هـــو أن اســـم سمة كلاً من frame وiframe وappears يظهر CDATA وليس كمعامل مرجع كينونة والسبب هو عدم وجود أي قيد واضح على الأطر غير أن يكونوا CDATA ولا تطبق أي لغة نظـــام أي شئ إلى CDATA في تلك الحالة

وحدة الارتباط النمطية

تعرف الوحدة النمطية التالية التي يستوردها كل من DTD الثابت والانتقالي وهـــي XHTML1-image.mod الموضحة في تعليمات برمجة ٢٠-٢١ العناصر الارتباط a وbase وlink.

تعليمات البرمجة ٢٠-٢١: XHTML1-image.mod وحدة الارتباط النمطية

```
<!- ................................................... ->
<!- XHTML 1.0 Linking Module  ............................ ->
<!- file: XHTML1-linking.mod
```

```
    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-linking.mod 1.13 99/04/01 SMI

    This DTD module is identified by the PUBLIC
    and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
    SYSTEM "XHTML1-linking.mod"

    Revisions:
# 1998-10-27  exclusion on 'a' within 'a' removed for XML
# 1998-11-15  moved shape and coords attributes on 'a' to
              csismap module
# 1999-01-14  moved onfocus and onblur attributes on 'a' to
              events module
    ................................ ->

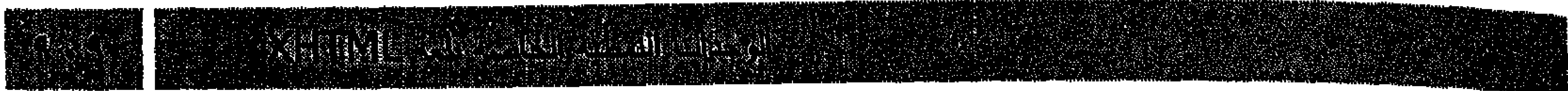
<!- d2. Linking

      a, base, link
->

<!- ........... Anchor Element ........... ->

<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA" >

<!ENTITY % A.content  "( #PCDATA | %Inline-noa.mix; )*" >
<!ELEMENT a  %A.content; >
<!ATTLIST a
     %Common.attrib;
     name        CDATA               #IMPLIED
     href        %URI;               #IMPLIED
     %Alink.attrib;
     charset     %Charset;           #IMPLIED
     type        %ContentType;       #IMPLIED
```



```
    hreflang    %LanguageCode;        #IMPLIED
    rel         %LinkTypes;           #IMPLIED
    rev         %LinkTypes;           #IMPLIED
    accesskey   %Character;           #IMPLIED
    tabindex    %Number;              #IMPLIED
>

<!- ........... Base Element ........... ->

<!ENTITY % Base.content "EMPTY" >
<!ELEMENT base %Base.content; >
<!ATTLIST base
    href        %URI;                 #REQUIRED
>

<!- ........... Link Element ........... ->

<!- Relationship values can be used in principle:

  a) for document specific toolbars/menus when used
     with the LINK element in document head e.g.
     start, contents, previous, next, index, end, help
  b) to link to a separate style sheet (rel=stylesheet)
  c) to make a link to a script (rel=script)
  d) by stylesheets to control how collections of
     html nodes are rendered into printed documents
  e) to make a link to a printable version of this document
     e.g. a postscript or pdf version
     (rel=alternate media=print)
->

<!ENTITY % Link.content "EMPTY" >
<!ELEMENT link %Link.content; >
<!ATTLIST link
    %Common.attrib;
    charset     %Charset;             #IMPLIED
    href        %URI;                 #IMPLIED
    hreflang    %LanguageCode;        #IMPLIED
```

```
    type        %ContentType;        #IMPLIED
    rel         %LinkTypes;          #IMPLIED
    rev         %LinkTypes;          #IMPLIED
    media       %MediaDesc;          #IMPLIED
>

<!- end of XHTML1-linking.mod ->
```

الوحدة النمطية لخريطة صور من جانب العميل

تعرف الوحدة النمطية التالية التي تستوردها كلا من DTD الثابت والانتقالي csismap.mod الموضحة في تعليمات برمجة ٢٠-٢٢ عناصر خريطة الصورة من جانب العميل map وarea ويذود عنصر map خريطة صورة من جانب العميل ويجب أن يحتوي على واحد أو أكثر من عناصر مستوى الكتلة والعناصر المتعددة أو عناصر ولعنصر area مجموعة سمات غير قياسية وغير عادية يختلف عنصر area عن باقي عناصر HTML فعنصر HTML فقط هو الذي يعمل مثل الرسم الاتجاهي.

تعليمات البرمجة ٢٠-٢٢: XHTML1-csismap.mod الوحدة النمطية لخريطة صورة من جانب العميل

```
<!- ..................................................... ->
<!- XHTML 1.0 Client-side Image Map Module  ............................. ->
<!- file: XHTML1-csismap.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-csismap.mod 1.15 99/04/01 SMI

    This DTD module is identified by the
    PUBLIC and SYSTEM identifiers:

 PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Maps//EN"
 SYSTEM "XHTML1-csismap.mod"
```

```
    Revisions:
# 1999-01-31  fixed map content model (errata)
    ................................................. ->

<!- d3.2. Client-side Image Maps

       area, map
->

<!- These can be placed in the same document or grouped in a
    separate document although this isn't widely supported ->

<!ENTITY % Map.content
  "(( %Heading.class; | %List.class; | %Block.mix; ) | area)+">
<!ELEMENT map  %Map.content; >
<!ATTLIST map
     %Common.attrib;
     name        CDATA                  #REQUIRED
>

<!ENTITY % Area.content  "EMPTY" >
<!ELEMENT area  %Area.content; >
<!ATTLIST area
     %Common.attrib;
     href         %URI;                 #IMPLIED
     shape        %Shape;               'rect'
     coords       %Coords;              #IMPLIED
     nohref       (nohref)              #IMPLIED
     alt          %Text;                #REQUIRED
     tabindex     %Number;              #IMPLIED
     accesskey    %Character;           #IMPLIED
>

<!- modify anchor (<a>) attribute definition list to
    allow for client-side image maps ->

<!ATTLIST a
     shape        %Shape;               'rect'
```

```
    coords      %Coords;            #IMPLIED
>

<!- end of XHTML1-csismap.mod ->
```

الوحدة النمطية لعنصر الكائن

الوحدة النمطية التالية التي يستوردها كلا من DTD الثابت والانتقالي والموضحة في تعليمات برمجة XHTML1-object.mod هي وحدة نمطية بسيطة تعرف عناصر object و param المستخدمة لتضمين محتويات لا تتبع HTML مثل تطبيقات Java وعناصر تحكم ActiveX وغيرها في صفحات الويب.

تعليمات البرمجة ٢٠-٢٣: XHTML1-object.mod الوحدة النمطية للكائن

```
<!- ................................................... ->
<!- XHTML 1.0 External Inclusion Module ............................... ->
<!- file: XHTML1-object.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-object.mod 1.16 99/04/01 SMI

    This DTD module is identified by the
    PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
    SYSTEM "XHTML1-object.mod"

    Revisions:
# 1999-01-31  changed object's archive attr
            to allow for multiple URIs
# 1999-01-31  corrected transitional attributes (errata)
    ................................................... ->

<!- d3.3. Objects
```

```
      object, param

    object is used to embed objects as part of HTML pages;
    param elements should precede other content.
->

<!ENTITY % Object.content  "( %Flow.mix; | param )*" >
<!ELEMENT object  %Object.content; >
<!ATTLIST object
     %Common.attrib;
     declare     (declare)            #IMPLIED
     classid     %URI;                #IMPLIED
     codebase    %URI;                #IMPLIED
     data        %URI;                #IMPLIED
     type        %ContentType;        #IMPLIED
     codetype    %ContentType;        #IMPLIED
     archive     %URIs;               #IMPLIED
     standby     %Text;               #IMPLIED
     height      %Length;             #IMPLIED
     width       %Length;             #IMPLIED
     usemap      %URI;                #IMPLIED
     name        CDATA                #IMPLIED
     tabindex    %Number;             #IMPLIED
>

<![%XHTML.Transitional;[
<!- additional Transitional attributes ->
<!ATTLIST object
     %IAlign.attrib;
     border      %Pixels;             #IMPLIED
     hspace      %Pixels;             #IMPLIED
vspace      %Pixels;             #IMPLIED
>
]]>

<!ENTITY % Param.content  "EMPTY" >
<!ELEMENT param  %Param.content; >
<!ATTLIST param
```

```
      id          ID                    #IMPLIED
      name        CDATA                 #REQUIRED
      value       CDATA                 #IMPLIED
      valuetype   (data|ref|object)     'data'
      type        %ContentType;         #IMPLIED
>

<!- end of XHTML1-object.mod ->
```

يتم تعريف عنصرين فقط هما aram وobject ونموذج المحتوى لعنصـــر object يظـهر باستخدام كينونات Flow.mix وparam ولا حظ أن نموذج المحتوى المختلط لعنصـــر object يتطلب تعريف أكثر دقة مما يستخدم في الوقت الحالي وهذا هو هدف وجـــود تعليـق param elements should precede other content ومع ذلك لا يمكن أن تحدد DTD أن عنـاصر param تنسيق أي محتوى أخر حيث يتطلب المحتوى المختلط أن تـــأتي #PCDATA أولا وأن يستخدم الاختيار بدلاً من التسلسل.

الوحدة النمطية لعنصر تطبيق Java صغير

اخترع عنصر Applet شركة Sun وذلك لتضمين تطبيقات Java في صفحات الويب والوحـــدة النمطية التالية التي يستوردها فقط DTD الانتقالي الموضحـــة فـــي تعليمـــات برمجـــة ٢٠-٢٤ XHTML1-applet.mod, هي وحدة نمطية تعرف عنصر applet وتقلل لغة HTML 4.0 مـــن حجم عنصر applet لصالح عناصر object العامة الأخرى والتي نستطيع تضمين ليس فقـــط التطبيقات الصغيرة ولكن أيضا عنـــاصر تحكـــم ActiveX والصـــور ورســـوم Shockwave المتحركة وكذلك أفلام QuickTime وأنواع أخرى من محتويات الوسائط المتعددة وبالتالي فـــإن XHTML DTD الانتقالي فقط يستخدم إحداث التطبيقات الصغيرة النمطية.

تعليمات البرمجة ٢٠-٢٤ : XHTML1-applet.mod الوحدة النمطية للتطبيق

```
<!- ................................................... ->
<!- XHTML 1.0 Draft Document Java Applet Module  ......................... -
>
<!- file: XHTML1-applet.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
```

```
    Reserved.
    Revision: @(#)XHTML1-applet.mod 1.14 99/04/01 SMI

    This DTD module is identified by the
    PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML V1.0 Java Applets//EN"
    SYSTEM "XHTML1-applet.mod"

    Revisions:
    (none)
    ........................................ ->

<!- d4. Scripting

      applet
->

<!- One of code or object attributes must be present.
    Place param elements before other content.
->

<!ENTITY % Applet.content  "( param | %Flow.mix; )*">
<!ELEMENT applet  %Applet.content; >
<!ATTLIST applet
    %Core.attrib;
    codebase    %URI;       #IMPLIED
    archive     CDATA       #IMPLIED
    code        CDATA       #IMPLIED
    object      CDATA       #IMPLIED
    alt         %Text;      #IMPLIED
    name        CDATA       #IMPLIED
    width       %Length;    #REQUIRED
    height      %Length;    #REQUIRED
    %IAlign.attrib;
    hspace      %Pixels;    #IMPLIED
    vspace      %Pixels;    #IMPLIED
>
```

```
<!- If the Object module that supplies the param element
    declarations is not used, redeclare %Param.local.module;
    as 'INCLUDE':  ->
<!ENTITY % Param.local.module  "IGNORE" >
<![%Param.local.module;[
<!ENTITY % Param.content  "EMPTY">
<!ELEMENT param  %Param.content; >
<!ATTLIST param
     id          ID                    #IMPLIED
     name        CDATA                 #REQUIRED
     value       CDATA                 #IMPLIED
     valuetype   (data|ref|object)     'data'
     type        %ContentType;         #IMPLIED
>
]]>

<!- end of XHTML1-applet.mod ->
```

يتشابه نموذج المحتوى وقائمة السمة لعنصر applet مع عنصر object وتوضح تعليمات برمجة ٢٢-٣ XHTML1-object.mod عنصر Param المستخدم لتمرير معاملات للتطبيقات وإذا حدث لأي سبب لم يتم استيراد ذلك فيمكن إعادة تعريف كينونة Param.local.module إلى NCLUDE بدلاً من IGNORE وتعرف DTD عنصر param.

الوحدة النمطية للقوائم

تعمل الوحدة النمطية الموضحة في تعليمات برمجة ٢٠-٢٥ XHTML1-list.mod في كلا من DTD وتعرف العناصر المستخدمة في القوائم سواء المرتبة أو غير المرتبة أو قوائم التعريف.

تعليمات البرمجة ٢٠-٢٥: XHTML-list-mod وحدة voyager النمطية للقوائم

```
<!- ............................................ ->
<!- XHTML 1.0 Lists Module  ............................................ ->
<!- file: XHTML1-list.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
```

```
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-list.mod 1.13 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
    SYSTEM "XHTML1-list.mod"

    Revisions:
    (none)
    .................................................. ->

<!- a3. Lists

      dl, dt, dd, ol, ul, li

    A conditional section includes additional declarations for
    the Transitional DTD

      dir, menu
->

<!- definition lists - DT for term, DD for its definition ->

<!ENTITY % Dl.content  "( dt | dd )+" >
<!ELEMENT dl  %Dl.content; >
<!ATTLIST dl
    %Common.attrib;
>

<!ENTITY % Dt.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT dt  %Dt.content; >
<!ATTLIST dt
    %Common.attrib;
>
```

```
<!ENTITY % Dd.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT dd  %Dd.content; >
<!ATTLIST dd
    %Common.attrib;
>

<!- Ordered Lists (ol) numbered styles ->

<!ENTITY % Ol.content  "( li )+" >
<!ELEMENT ol  %Ol.content; >
<!ATTLIST ol
    %Common.attrib;
>

<!- Unordered Lists (ul) bullet styles ->

<!ENTITY % Ul.content  "( li )+" >
<!ELEMENT ul  %Ul.content; >
<!ATTLIST ul
    %Common.attrib;
>

<!ENTITY % Li.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT li  %Li.content; >
<!ATTLIST li
    %Common.attrib;
>

<![%XHTML.Transitional;[
<!- Ordered lists (ol) Numbering style

   1   arabic numbers     1, 2, 3, ...
   a   lower alpha        a, b, c, ...
   A   upper alpha        A, B, C, ...
   i   lower roman        i, ii, iii, ...
I   upper roman        I, II, III, ...

   The style is applied to the sequence number which by
```

```
   default is reset to 1 for the first list item in
   an ordered list.
->

<!ENTITY % OlStyle "CDATA" >

<!ATTLIST ol
    type       %OlStyle;          #IMPLIED
    compact    (compact)          #IMPLIED
    start      %Number;           #IMPLIED
>

<!- Unordered Lists (ul) bullet styles ->
<!ENTITY % UlStyle "(disc|square|circle)" >

<!ATTLIST ul
    type       %UlStyle;          #IMPLIED
    compact    (compact)          #IMPLIED
>

<!ENTITY % Dir.content  "( li )+" >
<!ELEMENT dir  %Dir.content; >
<!ATTLIST dir
    %Common.attrib;
    compact    (compact)          #IMPLIED
>

<!ENTITY % Menu.content  "( li )+" >
<!ELEMENT menu  %Menu.content; >
<!ATTLIST menu
    %Common.attrib;
    compact    (compact)          #IMPLIED
>
]]>

<!- end of XHTML1-list.mod ->
```

يمكن تعريف القوائم المرتبة والغير المرتبة بنفس الطريقة وتحتوي كلا منها علـــى عنصـــر قائمة واحد ol or ul والتي قد تحتوي على عنصر قائمه واحد أو اكثر li وقد يكون لعنـــاصر ol وul سمات %Common.attrib القياسية لعنصر HTML وتشبه قائمة التعريف ذلك إلا انه يتـــم استخدام زوجي dl dt بدلاً من عناصر قائمة li.

وحدة النماذج النمطية

تغطي الوحدة النمطيـــة الموضحـــة فـــي تعليمـــات برمجـــة ٢٠-٢٦ XHTML1-form.mod والمستخدمة بواسطة نوعي DTD عنـــاصر نمـــاذج HTML القياســـي وهـــي عنـــاصر form وlabel, input, select, optgroup, option, textarea, fieldset, legend, and button وهي وحدة نمطية معقدة نوعاً وتعكس تعقد نماذج HTML.

تعليمات البرمجة ٢٠-٢٦: XHTML1-form.mod الوحدة النمطية لنماذج XHTML

```
<!- ............................................... ->
<!- XHTML 1.0 Forms Module ........................................... ->
<!- file: XHTML1-form.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-form.mod 1.18 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
    SYSTEM "XHTML1-form.mod"

    Revisions:
  # 1998-10-27  exclusion on form within form removed for XML
  # 1998-11-10  changed button content model to mirror exclusions
  # 1999-01-31  added 'accept' attribute on form (errata)
    ................................................. ->
```

```
<!- d7. Forms

        form, label, input, select, optgroup, option, textarea,
        fieldset, legend, button
->

<![%XHTML.Transitional;[
<!ENTITY % Form.content
    "( %Heading.class;
     | %List.class;
     | %Inline.class;
     | %Block-noform.mix;
     | fieldset )*"
>
]]>
<!ENTITY % Form.content
    "( %Heading.class;
     | %List.class;
| %Block-noform.mix;
     | fieldset )+"
>

<!ELEMENT form  %Form.content; >
<!ATTLIST form
%Common.attrib;
action        %URI;         #REQUIRED
method        (get|post)    'get'
enctype       %ContentType;  'application/x-www-form-urlencoded'
accept-charset %Charsets;   #IMPLIED
accept        %ContentTypes; #IMPLIED
>

<!- Each label must not contain more than ONE field ->

<!ENTITY % Label.content
   "( #PCDATA
    | %Inlstruct.class;
    | %Inlpres.class;
```

```
    | %Inlphras.class;
    | %Inlspecial.class;
    | input | select | textarea | button
    | %Misc.class; )*"
>
<!ELEMENT label  %Label.content; >
<!ATTLIST label
    %Common.attrib;
    for          IDREF                  #IMPLIED
    accesskey    %Character;            #IMPLIED
>

<!ENTITY % InputType.class
    "( text | password | checkbox | radio | submit
     | reset | file | hidden | image | button )"
>

<!- attribute name required for all but submit & reset ->

<!ENTITY % Input.content  "EMPTY" >
<!ELEMENT input  %Input.content; >
<!ATTLIST input
    %Common.attrib;
    type        %InputType.class;       'text'
    name         CDATA                  #IMPLIED
    value        CDATA                  #IMPLIED
    checked      (checked)              #IMPLIED
    disabled     (disabled)             #IMPLIED
    readonly     (readonly)             #IMPLIED
    size         CDATA                  #IMPLIED

<!- There are also 16 widely known color names with their sRGB values:

Black =#000000 Maroon =#800000 Green  = #008000 Navy =
#000080
Silver=#C0C0C0 Red    =#FF0000 Lime   = #00FF00 Blue = #0000FF
Gray  =#808080 Purple =#800080 Olive  = #808000 Teal = #008080
White =#FFFFFF Fuchsia=#FF00FF Yellow = #FFFF00 Aqua = #00FFFF
```

```
->

<!ATTLIST body
    bgcolor     %Color;          #IMPLIED
    text        %Color;          #IMPLIED
    link        %Color;          #IMPLIED
    vlink       %Color;          #IMPLIED
    alink       %Color;          #IMPLIED
    background  %URI;            #IMPLIED
>
]]>

<!ENTITY % Html.content "( head, body )" >

<!-version and namespace attribute values defined in driver->
<!ENTITY % Version.attrib
   "version     CDATA       #FIXED '%HTML.version;'" >
<!ENTITY % Ns.attrib
   "xmlns       %URI;       #FIXED '%XHTML.ns;'" >

<!ELEMENT html %Html.content; >
<!ATTLIST html
    %I18n.attrib;
    %Version.attrib;
    %Ns.attrib;
>

<!- end of XHTML1-struct.mod ->
```

اقتربت تلك الوحدة النمطية من حدود DTD وقد وضح في العديد من المرات تعليقات تحـــدد قواعد يصعب أو يستحيل تضمينها في التعريفات فمثلاً التعليـق attribute name required for all but submit & reset لعناصر input فيمكن تحديد أن يكون كل عناصر input سـمة nasme ولكن لا يمكن تحديد أن يحصل عليها البعض والبعض الاخر لا.

قد تعتقد أن هذا يشير إلى وجود عيب في لغة HTML وليس في DTD وقد يكون هذا صحيحاً فلا يلزم أن تكون أزرار إعادة التعيين "الاسترجاع" أو التنفيذ أه تكون عناصر input ومع ذلـــك فهناك العديد من الاماكن في هذه الوحدة النمطية التي يبدأ فيها DTD الانهيار تحت ثقلة والمـــراد

توضيحة هنا هو أن XML و DTD قد تمن تصميمهم لعرض مستندات ثابتة وليس الاستخدام التفاعلي.

وحدة الجدول النمطية

تعريف الوحدة النمطية التالية XHTML1-table.mod module الموضحة في تعليمات برمجة ٢٠-٢٧ والمستخدمة بواسطة كلاً من DTD العناصر المستخدمة لعرض الجداول في HTML وخاصة caption و col و colgroup و table و tbody و td و tfoot و th و thead و tr ومثل عناصر النموذج يجب أن تظهر معظم تلك العناصر داخل عنصر element فقط وبالتالي تعمل هذه الوحدة النمطية أطوال إلى حد ما حيث أنها لا تعتمد على عناصر معرفة من قبل وحيث أن العديد من العناصر المعرفة هنا لا تظهر في أي مكان أخر.

تعليمات البرمجة ٢٠-٢٧: XHTML1-table.mod وحدة الجداول XHTML النمطية

```
<!- ............................................... ->
<!- XHTML 1.0 Table Module ............................................ ->
<!- file: XHTML1-table.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-table.mod 1.15 99/04/01 SMI

    This DTD module is identified by the
    PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
    SYSTEM "XHTML1-table.mod"

    Revisions:
    (none)
    ....................................... ->
<!- d6. HTML 4.0 Tables
```

```
      caption, col, colgroup, table, tbody,
      td, tfoot, th, thead, tr

    A conditional section includes additional
    declarations for the Transitional DTD
->

<!- IETF HTML table standard, see [RFC1942] ->

<!- The border attribute sets the thickness of the frame
    around the table. The default units are screen pixels.

    The frame attribute specifies which parts of the frame
    around the table should be rendered. The values are not
    the same as CALS to avoid a name clash with the valign
    attribute.

    The value "border" is included for backwards compatibility
    with <table border> which yields frame=border and
    border=implied For <table border="1"> you get border="1"
    and frame="implied". In this case, it is appropriate to
    treat this as frame=border for backwards compatibility
    with deployed browsers.
->

<!ENTITY % TFrame
"(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!- The rules attribute defines which rules to draw between cells:

    If rules is absent then assume:

    "none" if border is absent or border="0" otherwise "all"
->

<!ENTITY % TRules "(none | groups | rows | cols | all)">

<!- horizontal placement of table relative to document ->
```

```
<!ENTITY % TAlign "(left|center|right)">

<!- horizontal alignment attributes for cell contents ->
<!ENTITY % CellHAlign.attrib
   "align       (left|center|right|justify|char) #IMPLIED
    char        %Character;              #IMPLIED
    charoff     %Length;                 #IMPLIED"
>

<!- vertical alignment attributes for cell contents ->
<!ENTITY % CellVAlign.attrib
   "valign      (top|middle|bottom|baseline) #IMPLIED"
>

<!ENTITY % CaptionAlign "(top|bottom|left|right)">

<!- Scope is simpler than axes attribute for common tables ->

<!ENTITY % Scope "(row|col|rowgroup|colgroup)" >

<!ENTITY % Table.content
    "( caption?, ( col* | colgroup* ),
     (( thead?, tfoot?, tbody+ ) | ( tr+ )))"
>
<!ELEMENT table  %Table.content; >
<!ATTLIST table
    %Common.attrib;
    summary      %Text;                  #IMPLIED
    width        %Length;                #IMPLIED
    border       %Pixels;                #IMPLIED
    frame        %TFrame;                #IMPLIED
    rules        %TRules;                #IMPLIED
    cellspacing  %Length;                #IMPLIED
    cellpadding  %Length;                #IMPLIED
    datapagesize CDATA                   #IMPLIED
>

<!ENTITY % Caption.content  "( #PCDATA | %Inline.mix; )*" >
```

```
<!ELEMENT caption  %Caption.content; >
<!ATTLIST caption
    %Common.attrib;
>

<!ENTITY % Thead.content  "( tr )+" >
<!ELEMENT thead  %Thead.content; >
<!- Use thead to duplicate headers when breaking table
   across page boundaries, or for static headers when
   TBODY sections are rendered in scrolling panel.

   Use tfoot to duplicate footers when breaking table
   across page boundaries, or for static footers when
   TBODY sections are rendered in scrolling panel.

   Use multiple tbody sections when rules are needed
   between groups of table rows.
->
<!ATTLIST thead
    %Common.attrib;
    %CellHAlign.attrib;
    %CellVAlign.attrib;
>

<!ENTITY % Tfoot.content  "( tr )+" >
<!ELEMENT tfoot  %Tfoot.content; >
<!ATTLIST tfoot
    %Common.attrib;
    %CellHAlign.attrib;
    %CellVAlign.attrib;
>

<!ENTITY % Tbody.content  "( tr )+" >
<!ELEMENT tbody  %Tbody.content; >
<!ATTLIST tbody
    %Common.attrib;
    %CellHAlign.attrib;
    %CellVAlign.attrib;
```

```
>

<!ENTITY % Colgroup.content  "( col )*" >
<!ELEMENT colgroup  %Colgroup.content; >
<!- colgroup groups a set of col elements. It allows you to
    group several semantically related columns together.
->
<!ATTLIST colgroup
     %Common.attrib;
     span         %Number;              '1'
     width        %MultiLength;         #IMPLIED
     %CellHAlign.attrib;
     %CellVAlign.attrib;
>

<!ENTITY % Col.content  "EMPTY" >
<!ELEMENT col  %Col.content; >
<!- col elements define the alignment properties for cells in
    one or more columns.

  The width attribute specifies the width of the columns, e.g.

      width="64"       width in screen pixels
      width="0.5*"      relative width of 0.5

    The span attribute causes the attributes of one
    col element to apply to more than one column.
  ->
<!ATTLIST col
     %Common.attrib;
     span         %Number;              '1'
     width        %MultiLength;         #IMPLIED
     %CellHAlign.attrib;
     %CellVAlign.attrib;
>

<!ENTITY % Tr.content  "( th | td )+" >
<!ELEMENT tr  %Tr.content; >
```

```
<!ATTLIST tr
    %Common.attrib;
    %CellHAlign.attrib;
    %CellVAlign.attrib;
>

<!- th is for headers, td for data, but for cells
   acting as both use td ->

<!ENTITY % Th.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT th  %Th.content; >
<!ATTLIST th
    %Common.attrib;
    abbr        %Text;             #IMPLIED
    axis        CDATA              #IMPLIED
    headers     IDREFS             #IMPLIED
    scope       %Scope;            #IMPLIED
    rowspan     %Number;           '1'
    colspan     %Number;           '1'
    %CellHAlign.attrib;
    %CellVAlign.attrib;
>

<!ENTITY % Td.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT td  %Td.content; >
<!ATTLIST td
    %Common.attrib;
    abbr        %Text;             #IMPLIED
    axis        CDATA              #IMPLIED
    headers     IDREFS             #IMPLIED
    scope       %Scope;            #IMPLIED
    rowspan     %Number;           '1'
    colspan     %Number;           '1'
    %CellHAlign.attrib;
    %CellVAlign.attrib;
>

<![%XHTML.Transitional;[
```

```
<!- additional Transitional attributes for XHTML tables:
   (in XML, multiple ATTLIST declarations are merged)
->

<!ATTLIST table
    align       %TAlign;              #IMPLIED
    bgcolor     %Color;               #IMPLIED
>

<!ATTLIST caption
    align       %CaptionAlign;        #IMPLIED
>

<!ATTLIST tr
bgcolor     %Color;                   #IMPLIED
>

<!ATTLIST th
    nowrap      (nowrap)              #IMPLIED
    bgcolor     %Color;               #IMPLIED
    width       %Pixels;              #IMPLIED
    height      %Pixels;              #IMPLIED
>

<!ATTLIST td
    nowrap      (nowrap)              #IMPLIED
    bgcolor     %Color;               #IMPLIED
    width       %Pixels;              #IMPLIED
    height      %Pixels;              #IMPLIED
>
]]>

<!- end of XHTML1-table.mod ->
```

الوحدة النمطية للبيانات التفصيلية

يستورد كلا من DTD الثابت والانتقالي الوحدة النمطية التالية XHTML1-meta.mod الموضحة في تعليمات برمجة ٢٠-٢٨ وتحصل على اسمها عن طريق تعريف عنصر meta الموضـــوع

في عناصر HTML هي head لتزويد الكلمة الأساسية والمؤلف والملخصـــات المســتخدمة وأي معلومات تبويب أخرى والتي تكون ذات فائدة لمستخدمي الويب وتعرف معلومات تفصيلية وفـــي تلك الحالة يكون اسم XHTML1-head.mod أفضل فيما عدا أن عنصر head لم يتم تعريفه.

تعليمات البرمجة ٢٠-٢٨ : XHTML1-meta.mod الوحدة النمطية لبيانات XHTML التفصيلية

```
<!- ............................................. ->
<!- XHTML 1.0 Document Metainformation Module .......................... -
>
<!- file: XHTML1-meta.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights
    Reserved.
    Revision: @(#)XHTML1-meta.mod 1.14 99/04/01 SMI

    This DTD module is identified by the PUBLIC and SYSTEM
    Identifiers:

    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
    SYSTEM "XHTML1-meta.mod"

    Revisions:
# 1998-11-11  title content model changed
          - exclusions no longer necessary
# 1999-02-01  removed isindex
    ........................................... ->

<!- d1. Meta Information

       meta, title
->

<!- The title element is not considered part of the flow of
    text. It should be displayed, for example as the page
    header or window title. Exactly one title is required per
```

```
    document.
->

<!ENTITY % Title.content  "( #PCDATA )" >
<!ELEMENT title  %Title.content; >
<!ATTLIST title
    %I18n.attrib;
>

<!ENTITY % Meta.content  "EMPTY" >
<!ELEMENT meta  %Meta.content; >
<!ATTLIST meta
    %I18n.attrib;
    http-equiv   NMTOKEN              #IMPLIED
    name         NMTOKEN              #IMPLIED
    content      CDATA                #REQUIRED
    scheme       CDATA                #IMPLIED
>

<!- end of XHTML1-meta.mod ->
```

الوحدة النمطية للهيكل

يضع آخر وحدة نمطية قياسية كل العناصر، السمات والكينونات المعرفة مسبقاً وتضمهم معاً في مستند HTML وهي وحدة XHTML1-struct.mod الموضحة في تعليمات برمجة ٢٠-٢٩ وبالتحديد فهي تعرف عناصر html وhead وbody.

تعليمات البرمجة ٢٠-٢٩: XHTML1-struct.mod وحدة هيكل XHTML النمطية

```
<!- ...................................................... ->
<!- XHTML 1.0 Structure Module ......................................... ->
<!- file: XHTML1-struct.mod

    This is XHTML 1.0, an XML reformulation of HTML 4.0.
    Copyright 1998-1999 W3C (MIT, INRIA, Keio),
```

```
All Rights Reserved.
Revision: @(#)XHTML1-struct.mod 1.15 99/04/01 SMI

This DTD module is identified by the PUBLIC and SYSTEM
identifiers:

PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
SYSTEM "XHTML1-struct.mod"

Revisions:
# 1998-10-27  content model on head changed to
          exclude multiple title or base
# 1998-11-11  ins and del inclusions on body removed,
          added to indiv. Elements
# 1998-11-15  added head element version attribute
          (restoring from HTML 3.2)
# 1999-03-24  %Profile.attrib; unused,
          but reserved for future use
 ................................................ ->

<!- a1. Document Structure

     body, head, html
->

<!ENTITY % Head-opts.mix  "( script | style | meta | link | object )*" >

<!ENTITY % Head.content "( title, base?, %Head-opts.mix; )" >

<!- reserved for future use with document profiles ->
<!ENTITY % Profile.attrib
   "profile     %URI;                #FIXED '%XHTML.profile;'" >

<!ELEMENT head  %Head.content; >
<!ATTLIST head
    %I18n.attrib;
    profile     %URI;                #IMPLIED
>
```

```
<![%XHTML.Transitional;[
<!- in Transitional, allow #PCDATA and inlines directly within
body ->

<!ENTITY % Body.content  "( #PCDATA | %Flow.mix; )*" >
]]>
<!ENTITY % Body.content
    "( %Heading.class;
     | %List.class;
     | %Block.class;
     | %Misc.class; )+"
>

<!ELEMENT body  %Body.content; >
<!ATTLIST body
     %Common.attrib;
>

<![%XHTML.Transitional;[
<!- ....  additional Transitional attributes on body .... ->

<!- There are also 16 widely known color names with their sRGB values:

Black =#000000 Maroon =#800000 Green  = #008000 Navy =
#000080
Silver=#C0C0C0 Red    =#FF0000 Lime   = #00FF00 Blue = #0000FF
Gray  =#808080 Purple =#800080 Olive  = #808000 Teal = #008080
White =#FFFFFF Fuchsia=#FF00FF Yellow = #FFFF00 Aqua = #00FFFF
->

<!ATTLIST body
     bgcolor      %Color;              #IMPLIED
     text         %Color;              #IMPLIED
     link         %Color;              #IMPLIED
     vlink        %Color;              #IMPLIED
     alink        %Color;              #IMPLIED
     background   %URI;                #IMPLIED
```

```
>
]]>

<!ENTITY % Html.content  "( head, body )" >

<!-version and namespace attribute values defined in driver->
<!ENTITY % Version.attrib
    "version      CDATA        #FIXED '%HTML.version;'" >
<!ENTITY % Ns.attrib
    "xmlns        %URI;        #FIXED '%XHTML.ns;'" >

<!ELEMENT html  %Html.content; >
<!ATTLIST html
     %I18n.attrib;
     %Version.attrib;
     %Ns.attrib;
>
<!- end of XHTML1-struct.mod ->
```

الوحدات النمطية غير القياسية

هناك عدد من الوحدات النمطية غير القياسية متضمنة في توزيع XHTML ولا تستخدم كجزء من تطبيق XHTML الرئيسي ولن يتم مناقشتها هنا ولكنها قد تكون جزء مفيد من برنامج المخصص ويتضمنها.

- XHTML1-form32.mod: نماذج HTML 3.2 "كمقابل لنماذج HTML 4.0 المستخدمة بواسطة XHTML".
- XHTML1-table32.mod: جداول HTML 3.2 "كمقابل لجداول HTML 4.0 المستخدمة في XHTML".
- XHTML1-math.mod: MathML مع بعض المواصفات البسيطة لجعلة متوافق تماماً مع XHTML.

مجموعات كينونة XHTML

يتطلب XML تعريف كل الكينونات "مع إمكانية استثناء مراجع الكينونات القياسية الخمسة، &lt و&gt و&apos و&quot و&amp". وتعرف XHTML DTD ثلاث مجموعات كينونات تعرف كافة الكينونات المستخدمة في HTML:

١- XHTML1-lat1.ent: الاحرف ١٦٠إلى ٢٥٥ من Latin-1 تعليمات برمجة ٢٠-٣٠.

٢- XHTML1-symbol.ent: تجميع الأحرف المفيده وعلامات الترقيم من خارج مجموعة Latin-1 مثل علامة Euro والواصلة الطويلة تعليمات برمجة ٢٠-٣١.

٣- XHTML1-special.ent: الحروف الهجائية اليونانية وتجميع الرموز الشائعة الاستخدام في الرياضيات مثل و تعليمات برمجة ٢٠-٣٢.

تتضمين مجموعات الكينونات هذه في كل إصدارات XHTML DTD وصولاً إلـــى الوحـــدة النمطية XHTML1-chars.mod. ولكل من مجموعات الكينونة هذه نفس تنسيق الأساسي.

١- تعليق يحتوي على العنوان أساسي والاستخدام ولذلك معلومات حقوق الطبع.

٢- العديد من تعريفات الكينونة الداخلية العامة وتعطي قيمة الكينونة العام كمرجع حرف إلى حرف Unicode وحيث أن لا يمكن أن يتذكر أحـــد كـــل احـــرف Unicode البالغـــه ٤٠٠٠٠ بالارقام فإن توصيف نص موجز للحرف إلذي يتم الرجوع آلية في تعليق يتبـع تعريف كينونة.

كينونات XHTML Latin-1

يعلن ملف XHTML1-lat1.ent الموضح في تعليمات برمجة ٢٠-٣٠ مراجع كينونات للنصــف الأعلى من ISO 8859-1 ومجموعة احرف Latin-1.

تعليمات البرمجة ٢٠-٣٠: XHTML1-lat1.ent: مجموعة كينونة XHTML للنصف الأعلى من وISO 8859-1 Latin-1

```
<!- XML-compatible ISO Latin 1 Character Entity Set for XHTML
1.0

    Typical invocation:

      <!ENTITY % XHTML1-lat1
         PUBLIC "-//W3C//ENTITIES Latin 1//EN//XML"
                "XHTML1-lat1.ent">
      %XHTML1-lat1;

    Revision:  @(#)XHTML1-lat1.ent 1.13 99/04/01 SMI
```

```
    Portions (C) International Organization for
    Standardization 1986 Permission to copy in any form is
    granted for use with conforming SGML systems and
    applications as defined in ISO 8879, provided this notice
    is included in all copies.
->
<!ENTITY nbsp "#&160;" ><!- no-break space=non-breaking space,
                          U+00A0 ISOnum ->
<!ENTITY iexcl  "#&161;" ><!- inverted exclamation mark,
                          U+00A1 ISOnum ->
<!ENTITY cent   "#&162;" ><!- cent sign,
                          U+00A2 ISOnum ->
<!ENTITY pound  "#&163;" ><!- pound sign,
                          U+00A3 ISOnum ->
<!ENTITY curren "#&164;" ><!- currency sign,
                          U+00A4 ISOnum ->
<!ENTITY yen    "#&165;" ><!- yen sign = yuan sign,
                          U+00A5 ISOnum ->
<!ENTITY brvbar "#&166;" ><!- broken bar =broken vertical bar,
                          U+00A6 ISOnum ->
<!ENTITY sect   "#&167;" ><!- section sign,
                          U+00A7 ISOnum ->
<!ENTITY uml    "#&168;" ><!- diaeresis = spacing diaeresis,
                          U+00A8 ISOdia ->
<!ENTITY copy   "#&169;" ><!- copyright sign,
                          U+00A9 ISOnum ->
<!ENTITY ordf   "#&170;" ><!- feminine ordinal indicator,
                          U+00AA ISOnum ->
<!ENTITY laquo  "#&171;" ><!- left-pointing double angle
quotation mark = left pointing guillemet,
                          U+00AB ISOnum ->
<!ENTITY not    "#&172;" ><!- not sign,
                          U+00AC ISOnum ->
<!ENTITY shy "#&173;" ><!- soft hyphen = discretionary hyphen,
                          U+00AD ISOnum ->
<!ENTITY reg    "#&174;" ><!- registered sign
                      = registered trade mark sign,
```

```
                              U+00AE ISOnum ->
<!ENTITY macr   "#&175;" ><!- macron = spacing macron
                         = overline = APL overbar,
                              U+00AF ISOdia ->
<!ENTITY deg    "#&176;" ><!- degree sign,
                              U+00B0 ISOnum ->
<!ENTITY plusmn "#&177;" ><!- plus-minus sign
                           = plus-or-minus sign,
                              U+00B1 ISOnum ->
<!ENTITY sup2   "#&178;" ><!- superscript two
                          = superscript digit two = squared,
                              U+00B2 ISOnum ->
<!ENTITY sup3   "#&179;" ><!- superscript three
                          = superscript digit three = cubed,
                              U+00B3 ISOnum ->
<!ENTITY acute  "#&180;" ><!- acute accent = spacing acute,
                              U+00B4 ISOdia ->
<!ENTITY micro  "#&181;" ><!- micro sign,
                              U+00B5 ISOnum ->
<!ENTITY para   "#&182;" ><!- pilcrow sign = paragraph sign,
                              U+00B6 ISOnum ->
<!ENTITY middot "#&183;" ><!- middle dot = Georgian comma
                              = Greek middle dot,
                              U+00B7 ISOnum ->
<!ENTITY cedil  "#&184;" ><!- cedilla = spacing cedilla,
                              U+00B8 ISOdia ->
<!ENTITY sup1   "#&185;" ><!- superscript one
                              = superscript digit one,
                              U+00B9 ISOnum ->
<!ENTITY ordm   "#&186;" ><!- masculine ordinal indicator,
                              U+00BA ISOnum ->
<!ENTITY raquo  "#&187;" ><!- right-pointing
  double angle quotation mark = right pointing guillemet,
                              U+00BB ISOnum ->
<!ENTITY frac14 "#&188;" ><!- vulgar fraction one quarter
                              = fraction one quarter,
                              U+00BC ISOnum ->
```



```
<!ENTITY frac12 "#&189;" ><!- vulgar fraction one half
                              = fraction one half,
                              U+00BD ISOnum ->
<!ENTITY frac34 "#&190;" ><!- vulgar fraction three quarters
                              = fraction three quarters,
                              U+00BE ISOnum ->
<!ENTITY iquest "#&191;" ><!- inverted question mark
                              = turned question mark,
                              U+00BF ISOnum ->
<!ENTITY Agrave "#&192;" ><!-latin capital letter A with grave
                         = latin capital letter A grave,
                              U+00C0 ISOlat1 ->
<!ENTITY Aacute "#&193;"><!-latin capital letter A with acute,
                              U+00C1 ISOlat1 ->
<!ENTITY Acirc  "#&194;" ><!- latin capital letter A
                         with circumflex,
                              U+00C2 ISOlat1 ->
<!ENTITY Atilde "#&195;"><!-latin capital letter A with tilde,
                              U+00C3 ISOlat1 ->
<!ENTITY Auml   "#&196;" ><!- latin capital letter A
                         with diaeresis,
                              U+00C4 ISOlat1 ->
<!ENTITY Aring  "#&197;" ><!- latin capital letter A
                         with ring above
                         = latin capital letter A ring,
                              U+00C5 ISOlat1 ->
<!ENTITY AElig  "#&198;" ><!- latin capital letter AE
                         = latin capital ligature AE,
                              U+00C6 ISOlat1 ->
<!ENTITY Ccedil "#&199;" ><!- latin capital letter C
                         with cedilla,
                              U+00C7 ISOlat1 ->
<!ENTITY Egrave "#&200;"><!-latin capital letter E with grave,
                              U+00C8 ISOlat1 ->
<!ENTITY Eacute "#&201;"><!-latin capital letter E with acute,
                              U+00C9 ISOlat1 ->
<!ENTITY Ecirc  "#&202;" ><!- latin capital letter E
```

```
                    with circumflex,
                         U+00CA ISOlat1 ->
<!ENTITY Euml   "#&203;" ><!- latin capital letter E with diaeresis,
                         U+00CB ISOlat1 ->
<!ENTITY Igrave "#&204;"><!-latin capital letter I with grave,
                         U+00CC ISOlat1 ->
<!ENTITY Iacute "#&205;"><!-latin capital letter I with acute,
                         U+00CD ISOlat1 ->
<!ENTITY Icirc  "#&206;" ><!- latin capital letter I
                    with circumflex,
                         U+00CE ISOlat1 ->
<!ENTITY Iuml   "#&207;" ><!- latin capital letter I
                    with diaeresis,
                         U+00CF ISOlat1 ->
<!ENTITY ETH    "#&208;" ><!- latin capital letter ETH,
                         U+00D0 ISOlat1 ->
<!ENTITY Ntilde "#&209;"><!-latin capital letter N with tilde,
                         U+00D1 ISOlat1 ->
<!ENTITY Ograve "#&210;"><!-latin capital letter O with grave,
                         U+00D2 ISOlat1 ->
<!ENTITY Oacute "#&211;"><!-latin capital letter O with acute,
                         U+00D3 ISOlat1 ->
<!ENTITY Ocirc  "#&212;" ><!- latin capital letter O
                    with circumflex,
                         U+00D4 ISOlat1 ->
<!ENTITY Otilde "#&213;"><!-latin capital letter O with tilde,
                         U+00D5 ISOlat1 ->
<!ENTITY Ouml   "#&214;" ><!- latin capital letter O
                    with diaeresis,
                         U+00D6 ISOlat1 ->
<!ENTITY times  "#&215;" ><!- multiplication sign,
                         U+00D7 ISOnum ->
<!ENTITY Oslash "#&216;"><!-latin capital letter O with stroke
                    = latin capital letter O slash,
                         U+00D8 ISOlat1 ->
<!ENTITY Ugrave "#&217;"><!-latin capital letter U with grave,
                         U+00D9 ISOlat1 ->
```

```
<!ENTITY Uacute "#&218;"><!-latin capital letter U with acute,
                          U+00DA ISOlat1 ->
<!ENTITY Ucirc  "#&219;" ><!- latin capital letter U
                       with circumflex,
                          U+00DB ISOlat1 ->
<!ENTITY Uuml   "#&220;" ><!- latin capital letter U
                       with diaeresis,
                          U+00DC ISOlat1 ->
<!ENTITY Yacute "#&221;"><!-latin capital letter Y with acute,
                          U+00DD ISOlat1 ->
<!ENTITY THORN  "#&222;" ><!- latin capital letter THORN,
                          U+00DE ISOlat1 ->
<!ENTITY szlig  "#&223;" ><!- latin small letter sharp s
                          = ess-zed,
                          U+00DF ISOlat1 ->
<!ENTITY agrave "#&224;" ><!- latin small letter a with grave
                        = latin small letter a grave,
                          U+00E0 ISOlat1 ->
<!ENTITY aacute "#&225;" ><!- latin small letter a with acute,
                          U+00E1 ISOlat1 ->
<!ENTITY acirc  "#&226;" ><!- latin small letter a
                       with circumflex,
                          U+00E2 ISOlat1 ->
<!ENTITY atilde "#&227;" ><!- latin small letter a with tilde,
                          U+00E3 ISOlat1 ->
<!ENTITY auml   "#&228;" ><!- latin small letter a
                       with diaeresis,
                          U+00E4 ISOlat1 ->
<!ENTITY aring  "#&229;" ><!- latin small letter a
                       with ring above
                         = latin small letter a ring,
                          U+00E5 ISOlat1 ->
<!ENTITY aelig  "#&230;" ><!- latin small letter ae
                          = latin small ligature ae,
                          U+00E6 ISOlat1 ->
<!ENTITY ccedil "#&231;" ><!- latin small letter c
                       with cedilla,
```

```
                        U+00E7 ISOlat1 ->
<!ENTITY egrave "#&232;" ><!- latin small letter e with grave,
                        U+00E8 ISOlat1 ->
<!ENTITY eacute "#&233;" ><!- latin small letter e with acute,
                        U+00E9 ISOlat1 ->
<!ENTITY ecirc  "#&234;" ><!- latin small letter e
                    with circumflex,
                        U+00EA ISOlat1 ->
<!ENTITY euml   "#&235;" ><!- latin small letter e
                    with diaeresis,
                        U+00EB ISOlat1 ->
<!ENTITY igrave "#&236;" ><!- latin small letter i with grave,
                        U+00EC ISOlat1 ->
<!ENTITY lacute "#&237;" ><!- latin small letter i with acute,
                        U+00ED ISOlat1 ->
<!ENTITY icirc  "#&238;" ><!- latin small letter i
                    with circumflex,
                        U+00EE ISOlat1 ->
<!ENTITY luml   "#&239;" ><!- latin small letter I
                    with diaeresis,
                        U+00EF ISOlat1 ->
<!ENTITY eth    "#&240;" ><!- latin small letter eth,
                        U+00F0 ISOlat1 ->
<!ENTITY ntilde "#&241;" ><!- latin small letter n with tilde,
                        U+00F1 ISOlat1 ->
<!ENTITY ograve "#&242;" ><!- latin small letter o with grave,
                        U+00F2 ISOlat1 ->
<!ENTITY oacute "#&243;" ><!- latin small letter o with acute,
                        U+00F3 ISOlat1 ->
<!ENTITY ocirc  "#&244;" ><!- latin small letter o
                    with circumflex,
                        U+00F4 ISOlat1 ->
<!ENTITY otilde "#&245;" ><!- latin small letter o with tilde,
                        U+00F5 ISOlat1 ->
<!ENTITY ouml   "#&246;" ><!- latin small letter o
                    with diaeresis,
                        U+00F6 ISOlat1 ->
```

```
<!ENTITY divide "#&247;" ><!- division sign,
                          U+00F7 ISOnum ->
<!ENTITY oslash "#&248;" ><!-latin small letter o with stroke,
                        = latin small letter o slash,
                          U+00F8 ISOlat1 ->
<!ENTITY ugrave "#&249;" ><!- latin small letter u with grave,
                          U+00F9 ISOlat1 ->
<!ENTITY uacute "#&250;" ><!- latin small letter u with acute,
                          U+00FA ISOlat1 ->
<!ENTITY ucirc  "#&251;" ><!- latin small letter u
                      with circumflex,
U+00FB ISOlat1 ->
<!ENTITY uuml   "#&252;" ><!- latin small letter u
                      with diaeresis,
                          U+00FC ISOlat1 ->
<!ENTITY yacute "#&253;" ><!- latin small letter y with acute,
                          U+00FD ISOlat1 ->
<!ENTITY thorn  "#&254;" ><!- latin small letter thorn with,
                          U+00FE ISOlat1 ->
<!ENTITY yuml   "#&255;" ><!- latin small letter y
                      with diaeresis,
                          U+00FF ISOlat1 ->
```

كينونات الاحرف الخاصة في XHTML

تعرف XHTML1-special.ent الموضحة في تعليمات برمجة ٢٠-٣١ الكينونات العامة المصنفة للأحرف ليس في Latin-1 ولكنها في Unicode

تعليمات البرمجة ٢٠-٣١: XHTML1-special تعريفات XHTML لكينونات عدة أحرف التي لا تتلائم في أي مكان آخر

```
<!-
 XML-compatible ISO Special Character Entity Set for XHTML 1.0

   Typical invocation:
```

```
    <!ENTITY % XHTML1-special
       PUBLIC "-//W3C//ENTITIES Special//EN//XML"
            "XHTML1-special.ent">
    %XHTML1-special;

   Revision:  @(#)XHTML1-special.ent 1.13 99/04/01 SMI

   Portions (C) International Organization for
   Standardization 1986: Permission to copy in any form is
   granted for use with conforming SGML systems and
   applications as defined in ISO 8879, provided this notice
   is included in all copies.
->

<!- Relevant ISO entity set is given unless names are newly
    introduced. New names (i.e., not in ISO 8879 list) do not
    clash with any existing ISO 8879 entity names. ISO 10646
    character numbers are given for each character, in hex.
    CDATA values are decimal conversions of the ISO 10646
    values and refer to the document character set. Names are
    Unicode 2.0 names.
->


<!- C0 Controls and Basic Latin ->
<!ENTITY quot "#&34;"> <!- quotation mark = APL quote,
                         U+0022 ISOnum ->
<!ENTITY amp  "#&38;"> <!- ampersand, U+0026 ISOnum ->
<!ENTITY lt   "#&60;"> <!- less-than sign, U+003C ISOnum->
<!ENTITY gt   "#&62;"> <!- greater-than sign, U+003E ISOnum->

<!- Latin Extended-A ->
<!ENTITY OElig   "#&338;"> <!-  latin capital ligature OE,
                         U+0152 ISOlat2 ->
<!ENTITY oelig   "#&339;"> <!-  latin small ligature oe,
                         U+0153 ISOlat2 ->
<!- ligature is a misnomer, this is a separate character
    in some languages ->
```

```
<!ENTITY Scaron  "#&352;"> <!-  latin capital letter S
                                with caron,
                                U+0160 ISOlat2 ->
<!ENTITY scaron  "#&353;"> <!-  latin small letter s
                                with caron,
                                U+0161 ISOlat2 ->
<!ENTITY Yuml    "#&376;"> <!-  latin capital letter Y
                                with diaeresis,
                                U+0178 ISOlat2 ->

<!- Spacing Modifier Letters ->
<!ENTITY circ    "#&710;"> <!-  modifier letter
                                circumflex accent,
                                U+02C6 ISOpub ->
<!ENTITY tilde   "#&732;"> <!-  small tilde, U+02DC ISOdia ->

<!- General Punctuation ->
<!ENTITY ensp    "#&8194;"> <!- en space, U+2002 ISOpub ->
<!ENTITY emsp    "#&8195;"> <!- em space, U+2003 ISOpub ->
<!ENTITY thinsp  "#&8201;"> <!- thin space, U+2009 ISOpub ->
<!ENTITY zwnj    "#&8204;"> <!- zero width non-joiner,
                                U+200C NEW RFC 2070 ->
<!ENTITY zwj     "#&8205;"> <!- zero width joiner,
                                U+200D NEW RFC 2070 ->
<!ENTITY lrm     "#&8206;"> <!- left-to-right mark,
                                U+200E NEW RFC 2070 ->
<!ENTITY rlm     "#&8207;"> <!- right-to-left mark,
                                U+200F NEW RFC 2070 ->
<!ENTITY ndash   "#&8211;"> <!- en dash, U+2013 ISOpub ->
<!ENTITY mdash   "#&8212;"> <!- em dash, U+2014 ISOpub ->
<!ENTITY lsquo   "#&8216;"> <!- left single quotation mark,
                                U+2018 ISOnum ->
<!ENTITY rsquo   "#&8217;"> <!- right single quotation mark,
                                U+2019 ISOnum ->
<!ENTITY sbquo   "#&8218;"> <!- single low-9 quotation mark,
                                U+201A NEW ->
```

```
<!ENTITY ldquo   "#&8220;"> <!- left double quotation mark,
                          U+201C ISOnum ->
<!ENTITY rdquo   "#&8221;"> <!- right double quotation mark,
                          U+201D ISOnum ->
<!ENTITY bdquo   "#&8222;"> <!- double low-9 quotation mark,
                          U+201E NEW ->
<!ENTITY dagger  "#&8224;"> <!- dagger, U+2020 ISOpub ->
<!ENTITY Dagger  "#&8225;"> <!- double dagger,
                          U+2021 ISOpub ->
<!ENTITY permil  "#&8240;"> <!- per mille sign,
                          U+2030 ISOtech ->
<!ENTITY lsaquo  "#&8249;"> <!- single left-pointing angle
                          quotation mark,
                          U+2039 ISO proposed ->
<!- lsaquo is proposed but not yet ISO standardized ->
<!ENTITY rsaquo  "#&8250;"> <!- single right-pointing
                          angle quotation mark,
                          U+203A ISO proposed ->
<!- rsaquo is proposed but not yet ISO standardized ->
<!ENTITY euro   "#&8364;"> <!-  euro sign, U+20AC NEW ->
```

كينونات رمز XHTML

تعريف XHTML1-symbol.ent الموضحة في تعليمات برمجة ٢٠-٣٢ الكينونات العامة الاحرف الهجائية اليونانية ورموز رياضية مختلفة مثل الإعداد الصحيحة وعلامات الجذر التربيعي.

تعليمات البرمجة ٢٠-٣٢: XHTML1-symbol.ent مجموعة كينونة Voyager للرموز الرياضية بما في ذلك الهجاء اليوناني

```
<!-- XML-compatible ISO Mathematical, Greek and Symbolic
   Character Entity Set for XHTML 1.0

   Typical invocation:
```

```
     <!ENTITY % XHTML1-symbol
        PUBLIC "-//W3C//ENTITIES Symbols//EN//XML"
             "XHTML1-symbol.ent">
     %XHTML1-symbol;

    Revision:  @(#)XHTML1-symbol.ent 1.13 99/04/01 SMI

    Portions (C) International Organization for
    Standardization 1986: Permission to copy in any form is
    granted for use with conforming SGML systems and
    applications as defined in ISO 8879, provided this notice
    is included in all copies.
-->

<!-- Relevant ISO entity set is given unless names are newly
    introduced. New names (i.e., not in ISO 8879 list) do not
    clash with any existing ISO 8879 entity names. ISO 10646
    character numbers are given for each character, in hex.
    CDATA values are decimal conversions of the ISO 10646
    values and refer to the document character set. Names are
    Unicode 2.0 names.
-->

<!-- Latin Extended-B -->
<!ENTITY fnof     "#&402;"> <!-- latin small f with hook
                          = function
                          = florin, U+0192 ISOtech>

<!-- Greek -->
<!ENTITY Alpha    "#&913;" ><!-- greek capital letter alpha,
                          U+0391 -->
<!ENTITY Beta     "#&914;" ><!-- greek capital letter beta,
                          U+0392 -->
<!ENTITY Gamma    "#&915;" ><!-- greek capital letter gamma,
                          U+0393 ISOgrk3 -->
<!ENTITY Delta    "#&916;" ><!-- greek capital letter delta,
                          U+0394 ISOgrk3 -->
```

```
<!ENTITY Epsilon  "#&917;" ><!– greek capital letter epsilon,
                           U+0395 –>
<!ENTITY Zeta     "#&918;" ><!– greek capital letter zeta,
                           U+0396 –>
<!ENTITY Eta      "#&919;" ><!– greek capital letter eta,
                           U+0397 –>
<!ENTITY Theta    "#&920;" ><!– greek capital letter theta,
                           U+0398 ISOgrk3 –>
<!ENTITY Iota     "#&921;" ><!– greek capital letter iota,
                           U+0399 –>
<!ENTITY Kappa    "&#922;" ><!– greek capital letter kappa,
                           U+039A –>
<!ENTITY Lambda   "#&923;" ><!– greek capital letter lambda,
                           U+039B ISOgrk3 –>
<!ENTITY Mu       "#&924;" ><!– greek capital letter mu,
                           U+039C –>
<!ENTITY Nu       "#&925;" ><!– greek capital letter nu,
                           U+039D –>
<!ENTITY Xi       "#&926;" ><!– greek capital letter xi,
                           U+039E ISOgrk3 –>
<!ENTITY Omicron  "#&927;" ><!– greek capital letter omicron,
                           U+039F –>
<!ENTITY Pi       "#&928;" ><!– greek capital letter pi,
                           U+03A0 ISOgrk3 –>
<!ENTITY Rho      "#&929;" ><!– greek capital letter rho,
                           U+03A1 –>
<!– there is no Sigmaf, and no U+03A2 character either –>
<!ENTITY Sigma    "#&931;" ><!– greek capital letter sigma,
                           U+03A3 ISOgrk3 –>
<!ENTITY Tau      "#&932;" ><!– greek capital letter tau,
                           U+03A4 –>
<!ENTITY Upsilon  "#&933;" ><!– greek capital letter upsilon,
                           U+03A5 ISOgrk3 –>
<!ENTITY Phi      "#&934;" ><!– greek capital letter phi,
```

```
                    U+03A6 ISOgrk3 –>
<!ENTITY Chi     "#&935;" ><!– greek capital letter chi,
                    U+03A7 –>
<!ENTITY Psi     "#&936;" ><!– greek capital letter psi,
                    U+03A8 ISOgrk3 –>
<!ENTITY Omega   "#&937;" ><!– greek capital letter omega,
                    U+03A9 ISOgrk3 –>
<!ENTITY alpha   "#&945;" ><!– greek small letter alpha,
                    U+03B1 ISOgrk3 –>
<!ENTITY beta    "#&946;" ><!– greek small letter beta,
                    U+03B2 ISOgrk3 –>
<!ENTITY gamma   "#&947;" ><!– greek small letter gamma,
                    U+03B3 ISOgrk3 –>
<!ENTITY delta   "#&948;" ><!– greek small letter delta,
                    U+03B4 ISOgrk3 –>
<!ENTITY epsilon "#&949;" ><!– greek small letter epsilon,
                    U+03B5 ISOgrk3 –>
<!ENTITY zeta    "#&950;" ><!– greek small letter zeta,
                    U+03B6 ISOgrk3 –>
<!ENTITY eta     "#&951;" ><!– greek small letter eta, U+03B7
                    ISOgrk3 –>
<!ENTITY theta   "#&952;" ><!– greek small letter theta,
                    U+03B8 ISOgrk3 –>
<!ENTITY iota    "#&953;" ><!– greek small letter iota,
                    U+03B9 ISOgrk3 –>
<!ENTITY kappa   "#&954;" ><!– greek small letter kappa,
                    U+03BA ISOgrk3 –>
<!ENTITY lambda  "#&955;" ><!– greek small letter lambda,
                    U+03BB ISOgrk3 –>
<!ENTITY mu      "#&956;" ><!– greek small letter mu, U+03BC
                    ISOgrk3 –>
<!ENTITY nu      "#&957;" ><!– greek small letter nu, U+03BD
                    ISOgrk3 –>
<!ENTITY xi      "#&958;" ><!– greek small letter xi, U+03BE
```

```
                              ISOgrk3 -->
<!ENTITY omicron  "#&959;" ><!-- greek small letter omicron,
                              U+03BF NEW -->
<!ENTITY pi       "#&960;" ><!-- greek small letter pi,
                              U+03C0 ISOgrk3 -->
<!ENTITY rho      "#&961;" ><!-- greek small letter rho,
                              U+03C1 ISOgrk3 -->
<!ENTITY sigmaf   "#&962;" ><!-- greek small letter final
                              sigma, U+03C2 ISOgrk3 -->
<!ENTITY sigma    "#&963;" ><!-- greek small letter sigma,
                              U+03C3 ISOgrk3 -->
<!ENTITY tau      "#&964;" ><!-- greek small letter tau,
                              U+03C4 ISOgrk3 -->
<!ENTITY upsilon  "#&965;" ><!-- greek small letter upsilon,
                              U+03C5 ISOgrk3 -->
<!ENTITY phi      "#&966;" ><!-- greek small letter phi,
                              U+03C6 ISOgrk3 -->
<!ENTITY chi      "#&967;" ><!-- greek small letter chi,
                              U+03C7 ISOgrk3 -->
<!ENTITY psi      "#&968;" ><!-- greek small letter psi,
                              U+03C8 ISOgrk3 -->
<!ENTITY omega    "#&969;" ><!-- greek small letter omega,
                              U+03C9 ISOgrk3 -->
<!ENTITY thetasym "#&977;" ><!-- greek small letter theta
                              symbol, U+03D1 NEW -->
<!ENTITY upsih    "#&978;" ><!-- greek upsilon with hook
                              symbol, U+03D2 NEW -->
<!ENTITY piv      "#&982;" ><!-- greek pi symbol,
                              U+03D6 ISOgrk3 -->

<!-- General Punctuation -->
<!ENTITY bull     "#&8226;" ><!-- bullet = black small circle,
                              U+2022 ISOpub  -->
<!-- bullet is NOT the same as bullet operator, U+2219 -->
```

```
<!ENTITY hellip   "#&8230;" ><!_ horizontal ellipsis
                  = three dot leader, U+2026 ISOpub _>
<!ENTITY prime    "#&8242;" ><!_ prime = minutes = feet,
                     U+2032 ISOtech _>
<!ENTITY Prime    "#&8243;" ><!_ double prime = seconds
                     = inches, U+2033 ISOtech _>
<!ENTITY oline    "#&8254;" ><!_ overline = spacing overscore,
                    U+203E NEW _>
<!ENTITY frasl    "#&8260;" ><!_ fraction slash, U+2044 NEW_>

<!_ Letterlike Symbols _>
<!ENTITY weierp   "#&8472;" ><!_ script capital P = power set
                   = Weierstrass p, U+2118 ISOamso _>
<!ENTITY image    "#&8465;" ><!_ blackletter capital I
                  = imaginary part, U+2111 ISOamso _>
<!ENTITY real     "#&8476;" ><!_ blackletter capital R
                 = real part symbol, U+211C ISOamso _>
<!ENTITY trade    "#&8482;" ><!_ trade mark sign,
                     U+2122 ISOnum _>
<!ENTITY alefsym  "#&8501;" ><!_ alef symbol
             = first transfinite cardinal, U+2135 NEW _>
<!_ alef symbol is NOT the same as hebrew letter alef,
   U+05D0 although the same glyph could be used to depict
   both characters _>

<!_ Arrows _>
<!ENTITY larr     "#&8592;" ><!_ leftwards arrow,
                     U+2190 ISOnum _>
<!ENTITY uarr     "#&8593;" ><!_upwards arrow, U+2191 ISOnum_>
<!ENTITY rarr     "#&8594;" ><!_ rightwards arrow,
                     U+2192 ISOnum _>
<!ENTITY darr     "#&8595;" ><!_ downwards arrow,
                     U+2193 ISOnum _>
<!ENTITY harr     "#&8596;" ><!_ left right arrow,
                     U+2194 ISOamsa _>
```

```
<!ENTITY crarr    "#&8629;" ><!_ downwards arrow with corner
                leftwards = carriage return, U+21B5 NEW _>
<!ENTITY lArr     "#&8656;" ><!_ leftwards double arrow,
                           U+21D0 ISOtech _>
<!_ Unicode does not say that lArr is the same as the
   'is implied by' arrow but also does not have any other
   character for that function. So ? lArr can
   be used for 'is implied by' as ISOtech suggests _>
<!ENTITY uArr     "#&8657;" ><!_ upwards double arrow,
                           U+21D1 ISOamsa _>
<!ENTITY rArr     "#&8658;" ><!_ rightwards double arrow,
                          U+21D2 ISOtech _>
<!_ Unicode does not say this is the 'implies' character
  but does not have  another character with this function
  so ? rArr can be used for 'implies' as ISOtech suggests _>
<!ENTITY dArr     "#&8659;" ><!_ downwards double arrow,
                           U+21D3 ISOamsa _>
<!ENTITY hArr     "#&8660;" ><!_ left right double arrow,
                          U+21D4 ISOamsa _>

<!_ Mathematical Operators _>
<!ENTITY forall   "#&8704;" ><!_ for all, U+2200 ISOtech _>
<!ENTITY part     "#&8706;" ><!_ partial differential,
                           U+2202 ISOtech  _>
<!ENTITY exist    "#&8707;"><!_there exists, U+2203 ISOtech_>
<!ENTITY empty    "#&8709;" ><!_ empty set = null set
                           = diameter, U+2205 ISOamso _>
<!ENTITY nabla    "#&8711;" ><!_ nabla = backward difference,
                          U+2207 ISOtech _>
<!ENTITY isin     "#&8712;" ><!_ element of, U+2208 ISOtech_>
<!ENTITY notin    "#&8713;" ><!_ not an element of,
                           U+2209 ISOtech _>
<!ENTITY ni       "#&8715;" ><!_ contains as member,
                           U+220B ISOtech _>
<!_ should there be a more memorable name than 'ni'? _>
```

```
<!ENTITY prod     "#&8719;" ><!_ n-ary product = product sign,
                      U+220F ISOamsb _>
<!_ prod is NOT the same character as U+03A0 'greek capital
    letter pi' though the same glyph might be used for both_>
<!ENTITY sum      "#&8721;" ><!_ n-ary sumation,
                      U+2211 ISOamsb _>
<!_ sum is NOT the same character as U+03A3
    'greek capital letter sigma' though the same glyph
    might be used for both _>
<!ENTITY minus    "#&8722;" ><!_ minus sign, U+2212 ISOtech_>
<!ENTITY lowast   "#&8727;" ><!_ asterisk operator,
                      U+2217 ISOtech _>
<!ENTITY radic    "#&8730;" ><!_ square root = radical sign,
                      U+221A ISOtech _>
<!ENTITY prop     "#&8733;" ><!_ proportional to,
                      U+221D ISOtech _>
<!ENTITY infin    "#&8734;" ><!_ infinity, U+221E ISOtech _>
<!ENTITY ang      "#&8736;" ><!_ angle, U+2220 ISOamso _>
<!ENTITY and      "#&8734;" ><!_ logical and = wedge,
                      U+2227 ISOtech _>
<!ENTITY or       "#&8744;" ><!_ logical or = vee,
                      U+2228 ISOtech _>
<!ENTITY cap      "#&8745;" ><!_ intersection = cap,
                      U+2229 ISOtech _>
<!ENTITY cup      "#&8746;" ><!_union = cup, U+222A ISOtech_>
<!ENTITY int      "#&8747;" ><!_ integral, U+222B ISOtech _>
<!ENTITY there4   "#&8756;" ><!_ therefore, U+2234 ISOtech _>
<!ENTITY sim      "#&8764;" ><!_ tilde operator
            = varies with = similar to, U+223C ISOtech _>
<!_ tilde operator is NOT the same character as the tilde,
    U+007E, although the same glyph might be used to
    represent both _>
<!ENTITY cong     "#&8773;" ><!_ approximately equal to, U+2245
ISOtech _>
<!ENTITY asymp    "#&8776;" ><!_ almost equal to
```

```
                          = asymptotic to, U+2248 ISOamsr _>
<!ENTITY ne        "#&8800;" ><!_ not equal to,
                              U+2260 ISOtech _>
<!ENTITY equiv     "#&8801;" ><!_ identical to,
                              U+2261 ISOtech _>
<!ENTITY le        "#&8804;" ><!_ less-than or equal to,
                              U+2264 ISOtech _>
<!ENTITY ge        "#&8805;" ><!_ greater-than or equal to,
                              U+2265 ISOtech _>
<!ENTITY sub       "#&8834;" ><!_ subset of, U+2282 ISOtech _>
<!ENTITY sup       "#&8835;" ><!_superset of, U+2283 ISOtech_>
<!_ note that nsup, 'not a superset of, U+2283' is not covered
    by the Symbol font encoding and is not included. Should it
    be, for symmetry? It is in ISOamsn  _>
<!ENTITY nsub      "#&8836;" ><!_ not a subset of,
                              U+2284 ISOamsn _>
<!ENTITY sube      "#&8838;" ><!_ subset of or equal to,
                              U+2286 ISOtech _>
<!ENTITY supe      "#&8839;" ><!_ superset of or equal to,
                              U+2287 ISOtech _>
<!ENTITY oplus     "#&8853;" ><!_ circled plus = direct sum,
                              U+2295 ISOamsb _>
<!ENTITY otimes    "#&8855;" ><!_ circled times
                          = vector product, U+2297 ISOamsb _>
<!ENTITY perp      "#&8869;" ><!_ up tack = orthogonal to
                          = perpendicular, U+22A5 ISOtech _>
<!ENTITY sdot      "#&8901;" ><!_ dot operator,
                              U+22C5 ISOamsb _>
<!_ dot operator is NOT the same character as
    U+00B7 middle dot _>

<!_ Miscellaneous Technical _>
<!ENTITY lceil     "#&8968;" ><!_ left ceiling = apl upstile,
                              U+2308 ISOamsc  _>
<!ENTITY rceil     "#&8969;" ><!_ right ceiling,
```

```
                        U+2309 ISOamsc _>
<!ENTITY lfloor  "#&8970;" ><!_ left floor = apl downstile,
                        U+230A ISOamsc _>
<!ENTITY rfloor  "#&8971;" ><!_ right floor,
                        U+230B ISOamsc _>
<!ENTITY lang    "#&9001;" ><!_ left-pointing angle bracket
                        = bra, U+2329 ISOtech _>
<!_ lang is NOT the same character as U+003C 'less than'
    or U+2039 'single left-pointing angle quotation mark' _>
<!ENTITY rang    "#&9002;" ><!_ right-pointing angle bracket
                        = ket, U+232A ISOtech _>
<!_ rang is NOT the same character as U+003E 'greater than'
    or U+203A 'single right-pointing angle quotation mark' _>

<!_ Geometric Shapes _>
<!ENTITY loz     "#&9674;" ><!_ lozenge, U+25CA ISOpub _>

<!_ Miscellaneous Symbols _>
<!ENTITY spades  "#&9824;" ><!_ black spade suit,
                        U+2660 ISOpub _>
<!_ black here seems to mean filled as opposed to hollow _>
<!ENTITY clubs   "#&9827;" ><!_ black club suit = shamrock,
                        U+2663 ISOpub _>
<!ENTITY hearts  "#&9829;" ><!_ black heart suit = valentine,
                        U+2665 ISOpub _>
<!ENTITY diams   "#&9830;" ><!_ black diamond suit,
                        U+2666 ISOpub _>
```

مجموعات DTD الإضافية المبسطة

لا تحتاج كل الأنظمة التي تعتمد على HTML على كل جزء موجود في HTML واعتماداً علـى احتياجات المستخدم قد تستطيع حذف نماذج وتطبيقات وصور وخريطة الصور ومزايات متطورة وتفاعلية في HTML فمثلاً ليرجع إلى مثال كره البيسبول في الجزء الأول من هذا الكتـاب فـإذا كنت ستعطي كل PLAYER عنصر BIO يمكن استخدام HTML بسيطة لتضمين نص أساسي مع كل لاعب.

الوحدات النمطية الأساسية التي سترغب في تضمينها في أي تطبيق يتم تصميمـــة باســتخدام XHTML هم.

- XHTML1-attribs.mod
- XHTML1-blkphras.mod
- XHTML1-blkpres.mod
- XHTML1-blkstruct.mod
- XHTML1-charent.mod
- XHTML1-inlphras.mod
- XHTML1-inlpres.mod
- XHTML1-inlstruct.mod
- XHTML1-model.mod
- XHTML1-names.mod

بالإضافة إلى ذلك فمن السهل دمج واحدات نمطية أخرى لهذة المجموعة الأساسية يمكن دمج XHTML1-image للصور وXHTML1-linking للنص المترابط بينما يمكن أن ترتبطهما فـــي DTD باستخدام مراجع معاملات كينونات خارجية كما يوضح مثال في الفصل ٢٣ ومن الســـهل حزف اختيار الأجزاء المطلوبة وغير المطلوبه هي نسخ إما DTD باستخدام مراجـــع معـــاملات كينونات خارجية كما يوضح مثال في الفصل ٢٣ والسهل حزف اختيار الأجزاء المطلوبة وغـــير المطلوب هي نسيخ إما DTD الثابت أو الانتقالي وIGNORE تجاهل الأجزاء التـــــي لا تريدهـــا تعليمات برمجة ٢٠-٣٣ هي نسخة من DTD الثابت لتعليمات برمجة ٢٠-٢١ حيث تتضمن فقط الوحدات النمطية المدرجة فيما سبق.

تعليمات البرمجة ٢٠-٣٣: DTD أساسية تدعم HTML الأساسي

```
<!- ................................................. ->
<!- Basic HTML for Player BIOs, based on XHTML 1.0 strict ->
<!- file: XHTML1-bb.dtd
->

<!- This derived from
    XHTML 1.0, an XML reformulation of HTML 4.0.

  Copyright 1998-1999 World Wide Web Consortium
```

```
(Massachusetts Institute of Technology, Institut National de
 Recherche en Informatique et en Automatique, Keio University).
 All Rights Reserved.

   Permission to use, copy, modify and distribute the XHTML
   1.0 DTD and its accompanying documentation for any purpose
   and without fee is hereby granted in perpetuity, provided
   that the above copyright notice and this paragraph appear
   in all copies. The copyright holders make no representation
   about the suitability of the DTD for any purpose.

   It is provided "as is" without expressed or implied
   warranty.

  Original Author:    Murray M. Altheim <altheim@eng.sun.com>
  Original Revision:  @(#)XHTML1-s.dtd 1.14 99/04/01 SMI

   The DTD is an XML variant based on the
   W3C HTML 4.0 DTD:

      Draft:     $Date: 1999/04/02 14:27:27 $

      Authors:    Dave Raggett <dsr@w3.org>
                  Arnaud Le Hors <lehors@w3.org>
                  Ian Jacobs <ij@w3.org>

->

<!- The version attribute has historically been a container
    for the DTD's public identifier (an FPI), but is unused
    in Strict:  ->
<!ENTITY % HTML.version  "" >
<!ENTITY % Version.attrib "" >

<!- The xmlns attribute on <html> identifies the
    default namespace to namespace-aware applications:  ->
<!ENTITY % XHTML.ns  "http://www.w3.org/TR/1999/REC-html-in-xml"
>
```

```
<!- reserved for future use with document profiles ->
<!ENTITY % XHTML.profile  "" >

<!- used to ignore Transitional features within modules ->
<!ENTITY % XHTML.Transitional    "IGNORE" >


<!- XHTML Base Architecture Module (optional) ......... ->
<!ENTITY % XHTML1-arch.module "IGNORE" >
<![%XHTML1-arch.module;[
<!ENTITY % XHTML1-arch.mod
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"
          "XHTML1-arch.mod" >
%XHTML1-arch.mod;
]]>

<!- Common Names Module ............................... ->
<!ENTITY % XHTML1-names.module "INCLUDE" >
<![%XHTML1-names.module;[
<!ENTITY % XHTML1-names.mod
   PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
          "XHTML1-names.mod" >
%XHTML1-names.mod;
]]>

<!- Character Entities Module ......................... ->
<!ENTITY % XHTML1-charent.module "INCLUDE" >
<![%XHTML1-charent.module;[
<!ENTITY % XHTML1-charent.mod
   PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
          "XHTML1-charent.mod" >
%XHTML1-charent.mod;
]]>

<!- Intrinsic Events Module ........................... ->
<!ENTITY % XHTML1-events.module "IGNORE" >
<![%XHTML1-events.module;[
```

```
<!ENTITY % XHTML1-events.mod
    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
          "XHTML1-events.mod" >
%XHTML1-events.mod;
]]>

<!- Common Attributes Module ......................... ->
<!ENTITY % XHTML1-attribs.module "INCLUDE" >
<![%XHTML1-attribs.module;[
<!ENTITY % align "" >
<!ENTITY % XHTML1-attribs.mod
    PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"
          "XHTML1-attribs.mod" >
%XHTML1-attribs.mod;
]]>

<!- Document Model Module ............................ ->
<!ENTITY % XHTML1-model.module "INCLUDE" >
<![%XHTML1-model.module;[
<!ENTITY % XHTML1-model.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"
          "XHTML1-model.mod" >
%XHTML1-model.mod;
]]>

<!- Inline Structural Module .......................... ->
<!ENTITY % XHTML1-inlstruct.module "INCLUDE" >
<![%XHTML1-inlstruct.module;[
<!ENTITY % XHTML1-inlstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
          "XHTML1-inlstruct.mod" >
%XHTML1-inlstruct.mod;
]]>

<!- Inline Presentational Module ...................... ->
<!ENTITY % XHTML1-inlpres.module "INCLUDE" >
<![%XHTML1-inlpres.module;[
<!ENTITY % XHTML1-inlpres.mod
```

```
   PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
        "XHTML1-inlpres.mod" >
%XHTML1-inlpres.mod;
]]>

<!- Inline Phrasal Module ............................ ->
<!ENTITY % XHTML1-inlphras.module "INCLUDE" >
<![%XHTML1-inlphras.module;[
<!ENTITY % XHTML1-inlphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
         "XHTML1-inlphras.mod" >
%XHTML1-inlphras.mod;
]]>

<!- Block Structural Module .......................... ->
<!ENTITY % XHTML1-blkstruct.module "INCLUDE" >
<![%XHTML1-blkstruct.module;[
<!ENTITY % XHTML1-blkstruct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
         "XHTML1-blkstruct.mod" >
%XHTML1-blkstruct.mod;
]]>

<!- Block Presentational Module ...................... ->
<!ENTITY % XHTML1-blkpres.module "INCLUDE" >
<![%XHTML1-blkpres.module;[
<!ENTITY % XHTML1-blkpres.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
         "XHTML1-blkpres.mod" >
%XHTML1-blkpres.mod;
]]>

<!- Block Phrasal Module ............................. ->
<!ENTITY % XHTML1-blkphras.module "INCLUDE" >
<![%XHTML1-blkphras.module;[
<!ENTITY % XHTML1-blkphras.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
         "XHTML1-blkphras.mod" >
```

```
%XHTML1-blkphras.mod;
]]>

<!- Scripting Module ................................. ->
<!ENTITY % XHTML1-script.module "IGNORE" >
<![%XHTML1-script.module;[
<!ENTITY % XHTML1-script.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
          "XHTML1-script.mod" >
%XHTML1-script.mod;
]]>

<!- Stylesheets Module ............................... ->
<!ENTITY % XHTML1-style.module "IGNORE" >
<![%XHTML1-style.module;[
<!ENTITY % XHTML1-style.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"
          "XHTML1-style.mod" >
%XHTML1-style.mod;
]]>

<!- Image Module ..................................... ->
<!ENTITY % XHTML1-image.module "IGNORE" >
<![%XHTML1-image.module;[
<!ENTITY % XHTML1-image.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
          "XHTML1-image.mod" >
%XHTML1-image.mod;
]]>

<!- Frames Module .................................... ->
<!ENTITY % XHTML1-frames.module  "IGNORE" >
<![%XHTML1-frames.module;[
<!ENTITY % XHTML1-frames.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
          "XHTML1-frames.mod" >
%XHTML1-frames.mod;
]]>
```

```
<!- Linking Module ................................... ->
<!ENTITY % XHTML1-linking.module "IGNORE" >
<![%XHTML1-linking.module;[
<!ENTITY % XHTML1-linking.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
           "XHTML1-linking.mod" >
%XHTML1-linking.mod;
]]>

<!- Client-side Image Map Module ...................... ->
<!ENTITY % XHTML1-csismap.module "IGNORE" >
<![%XHTML1-csismap.module;[
<!ENTITY % XHTML1-csismap.mod
  PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"
        "XHTML1-csismap.mod" >
%XHTML1-csismap.mod;
]]>

<!- Object Element Module ............................ ->
<!ENTITY % XHTML1-object.module "IGNORE" >
<![%XHTML1-object.module;[
<!ENTITY % XHTML1-object.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
           "XHTML1-object.mod" >
%XHTML1-object.mod;
]]>

<!- Lists Module .................................... ->
<!ENTITY % XHTML1-list.module "IGNORE" >
<![%XHTML1-list.module;[
<!ENTITY % XHTML1-list.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
           "XHTML1-list.mod" >
%XHTML1-list.mod;
]]>

<!- Forms Module .................................... ->
```

```
<!ENTITY % XHTML1-form.module "IGNORE" >
<![%XHTML1-form.module;[
<!ENTITY % XHTML1-form.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
          "XHTML1-form.mod" >
%XHTML1-form.mod;
]]>

<!- Tables Module ...................................... ->
<!ENTITY % XHTML1-table.module "IGNORE" >
<![%XHTML1-table.module;[
<!ENTITY % XHTML1-table.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
          "XHTML1-table.mod" >
%XHTML1-table.mod;
]]>

<!- Document Metainformation Module .................. ->
<!ENTITY % XHTML1-meta.module "IGNORE" >
<![%XHTML1-meta.module;[
<!ENTITY % XHTML1-meta.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
          "XHTML1-meta.mod" >
%XHTML1-meta.mod;
]]>

<!- Document Structure Module ......................... ->
<!ENTITY % XHTML1-struct.module "IGNORE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
    PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
          "XHTML1-struct.mod" >
%XHTML1-struct.mod;
]]>

<!- end of XHTML 1.0 Strict DTD  ......................... ->
<!- ................................................... ->
```

بعيداً عن بعض التغيرات في التعليقات في البداية للإشارة إلى أن هذه إصـــدار مشـــتق مـــن XHTML وبالتحديدDTD الثابت والتغيرات الوحيدة هي استبدال INCLUDE بكلمة IGNORE في العديد من مراجع معاملات الكينونات مثل XHTML1-struct.module.

وأصبح من الممكن إلغاء الأجزاء غير الضرورية نهائياً بدلاً من تجاهلها ويجعل هذا الإتجـــاة من السهل تضمينهم بسهولة إذا أرادت استخدامهم فيما بعد.

لا يمكن أن تسمي التطبيق الناتج HTML ولكن يقدم أسلوب إضافة هيكل نص مترابط أساسي إلى مجال DTD اكثر تحديداً. دون أخذ كل الوسائط المتعددة وهي HTML 4.0.

مثال لذلك بإضافة تعليمات برمجة ٢٠-٣٣ إلى DTD للاعب كرة البيسيبول من الفصل ١٠ يمكن إعطاء كل لاعب عنصر BIOGRAPHY يحتوي على HTML أساسي ويبدو التعريف كمـــا يلي:

```
<!ENTITY % XHTML1-bb.dtd SYSTEM "XHTML1-bb.dtd">

%XHTML1-bb.dtd;
<!ENTITY % BIOGRAPHY.content  "( #PCDATA | %Flow.mix; )*" >
<ELEMENT BIOGRAPHY %BIOGRAPHY.content;>
```

يمكن أن تحتوي BIOGRAPHY على أي شئ يمكن أن تحتوي كتلة HTML كما تم تعريـــف ذلك بواسطة وحدات XHTML النمطية المستخدمة هنا ويمكن استخدام أي من العناصر الأخـــرى أو مراجع كينونة نموذج المحتوى من وحدات XHTML النمطية.

ملاحظات حقوق الطبع في DTD

يمكن في حالة تصميم DTD فقط من أجل الاستخدام الشخصي أي لوضعة على موقـــع ويب خاص بالمصمم أو للتوثيق المطبوع داخل شركة وضـــع أي ملاحظـــات بشـــأن حقوق الطبع أما في حالة تصميم DTD لصناعة محددة أو المنطقة دراســـة بأكملـــها يجب مراعاة حقوق الطبع التي تكتب فمثـــلاً ملاحظـــة حقـــوق طبـــع بســـيطة مثـــل Copyright 1999 Elliotte Rusty Harold تجعل DTD إلى خادم ويب مختلـــف أو إلى مستند جديد دون تصريح واضح بذلك.

الهدف الأساسي في لغة XML هو إتاحة مستندات قياسية واسعة الانتشار واللوصـــول إلى هذا الهدف يجب لأي لغة ترميز يتم إنشاؤها سواء وصفت فـــي DTD أو DCD أو DDML أو DocumentD أو أي شئ أخر أن يتم استخدامها وطباعتها دون ضـــرورة الحصول على تصريح مسبق وأفضل ما يجب فعله هو وضع DTD في المجال العـــام

ملاحظات حقوق الطبع في DTD

لأنها أبسط وأسهل في الشرح للمحامين وتعمل المصادر المفتوحة بصورة جيدة أيضا وتعتبر عبارة حق الطبع التي تتيح إعادة الاستخدام ولكن لا تتيح تعديلات مناسبة في العديد من الأوجة.

لذا يجب التفكير بجدية في حقوق الطبع التي توضح في DTD فيجب أن تسأل نفسك قبل وضعها بعض الأسئلة مثل ما معنى تلك الحقوق؟ ما المراد من استخدام DTD؟ هل تحقيق عبارة حقوق الطبع المستخدمة هذا الغرض؟ فلن يتحقق الكثير من كتابة DTD لتنهجها صناعة ثم يغير قصد تمنع تلك الصناعة من انتهاجها بسبب ملاحظات حقوق الطبع.

بالرغم من أن هذا الكتاب بما فيه له حقوق طبع فقد تم وضع أمثلة التعليمات البرمجية على المجال العام وللقارئ حرية استخدام جزء أو كل تعليمات DTD البرمجية بإذن أو بدون الحصول على أذن.

تقليد التقنيات

نقل عن الرسام الشهر بابلوبيكاسو قوله أن الرسام الجيد ينقل بينما الرسام العظيم يسرق وكما وضح في أثناء استخدام هذا الكتاب فإن جزء من سبب أن XHTML DTD نمطية أي تمكن تقسيمها إلى أجزاء عديدة هو الاقتباس منها فإذا أراد المستخدم تنسيق نص مترابط أساسي كجزء من تطبيق XML الذي تقوم بتطويره فلا حاجة إلى اختراع واحد يمكن استيراد الوحدات النمطية المطلوبة ولهذا ميزة اضافية هو أن ناشري المستند الذين سيستخدموا تطبيق XML يكونوا متعادين على الترميز من لغة HTML وما يلي هو بعض النقيات التي يمكن استعارتها من XHTML DTD إلى DTD دون أخذ DTD نفسها.

التعليقات

تعتبر XHTML DTD واخره بالتعليقات ولكل ملف تعليق يعطي عنوان وحقوق الطبع المتصلة بالملف وكذلك جزء من محتويات الملفا قبل ظهور تعهريف واحد ويفصل كل جزء من الملف بواسطة تعليق جديد يحدد الهدف من الجزء ويوضح كل تعريف يناقش معنى التعريف وهذا يسهل قراءة وفهم الملف.

لكن مازال كل هذا لا يصل إلى درجة الكمال فالعديد من تعريفات السمات لا يتم التعليق عليها بصورة كافية ومثال لذلك هذا تعريف من XHTM1-applet.mod.

```
%Core.attrib;
```

```
  codebase    %URI;              #IMPLIED
  archive     CDATA              #IMPLIED
  code        CDATA              #IMPLIED
  object      CDATA              #IMPLIED
  alt         %Text;             #IMPLIED
  name        CDATA              #IMPLIED
  width       %Length;           #REQUIRED
  height      %Length;           #REQUIRED
  %IAlign.attrib;
  hspace      %Pixels;           #IMPLIED
  vspace      %Pixels;           #IMPLIED
>
```

لا توجد ما يشير إلى ما هي قيمة هذه السمات ويكون تعليق إضافي كالتالي مفيد في ذلك.

```
<!- ATTLIST applet
  codebase  the URI where of the directory from which the
            applet is downloaded; defaults to the URI of the
            document containing the applet tag
  archive   the name of the JAR file that contains the applet;
            omitted if the applet isnصt stored in a JAR
            archive
  code      the name of the main class of the applet
  object    the name of the serialized object that contains
            the main applet class; must match the name of the
            class in the applet attribute
  alt       text displayed if the applet cannot be located
  name      the name of the applet
  width     width of the applet in pixels
  height    height of the applet in pixels
  align     bottom, middle, top, left, or right
            meaning the bottom, middle, or top of the applet
            is aligned with the baseline or that the
            applet floats to the left or the right
  hspace    number of pixels with which
            to pad the left and right sides of the applet
  vspace    number of pixels with which
            to pad the top and bottom of the applet
->
```

بالطبع يمكن ايجاد كل ذلك بقراءة مواصفات HTML 4.0 وبالطبع ففي العديد مـــن المـــرات عندما تم ترك التوثيق الكامل لمستند نص أخر فلم تكتب لن يضير إضافة تعليق إضافية عند كتابة DTD لأول مرة.

يمكن جزء من المشكله في أن القيود على قيم السمات لا يتم التعبير عنها فـــي DTD فمثـــلاً يجب أن يكون كلاً من height وwidth إعداد صحيحة وفي المستقبل يمكـــن مواجهـــة تلـــك المشكلة عن طريق استخدام لغة نظام تكتب أعلى بناء جملة XML القياسي.

في حالات السمات المعقدة وتعريفات العناصر فمن المفيد تقديم مثال في تعليق.

```
<!-

  <applet width="500" height="500"
      codebase="http://www.site.com/directory/subdirectory/"
      archive="MyApplet.jar"
      code="MyApplet.class"
      object="MyApplet.ser"
      name="FirstInstance"
      align="top"
      hspace="5"
      vspace="5"
   >
    <param name="name1" value="value1"/>
    <param name="name2" value="value2"/>
    Some text for browsers that don't understand the
    applet tag
   </applet>

->
```

كينونات المعاملات

تستخدم XHTML DTD بصورة مكثفة كينونات المعاملات الداخلية والخارجية ويمكـــن أن تفـــي DTD بالغرض أيضا. ولمعاملات الكينونات العديد من الاستخدامات والتـــي تـــم شـــرحها فـــي XHTML DTD ويمكن استخدامها في:

- تقسيم نماذج المحتوي الطويلة وقوائم السمات إلى أجزاء متصلة يمكن التعامل بها.
- جعل مجموعات العناصر والسمات المشتركة قياسية.

- إتاحه تغير نماذج المحتوى وقوائم السمات لمختلف DTD.
- تحسين المستندات ونماذج المحتوى.
- ضغط DTD عن طريق إعاده استخدام النتالي العادي للنص.
- تقسيم DTD إلى وحدات نمطية أحادية متصلة.

تقسيم نماذج المحتوى الطويلة وقوائم السمات إلى اجزاء متصلة يمكن التعامل بها

يمكن أن يكون لعنصر HTML عادي مثل p على ٣٠ سمة أو اكثر والعديد من التوابع المحتملة ولذلك يصعب سردهم بالكامل في نموذج محتوي أو قائمة سمات ويمكن وضع السمات والعناصر المتصلة في مجموعات فمن الأفضل فصلهم في معاملات كينونات متعددة وما يلي هو تعريف عنصر XHTML لعنصر p.

```
<!ELEMENT p  %P.content; >
```

تستخدم مرجع معامل كينونة واحد بدلاً من اسماء العناصر منفصلة التي يلجأ إليها المرجع. وقائمة سمة HTML لعنصر P هي:

```
Here's XHTML's attribute list for p:
<!ATTLIST p
    %Common.attrib;
>
```

تستخدم كينونة معامل واحد بدلاً من أسماء المنفصلة وأنواع المحتوى التي تلجأ اليها.

جعل مجموعات العناصر والسمات المشتركة قياسية

عندالتعامل مع ٣٠ عنصر أو اكثر في قائمة فمن السهل تناسي عنصر إذا تطلب الافر تكرار القائمة فمثلاً قد يكون لمعظم عناصر HTML هذه السمات.

```
id class style title lang xml:lang dir onclick ondblclick
onmousedown onmouseup onmousemove onmouseout onkeypress
onkeydown onkeyup onclick ondblclick onmousedown onmouseup
onmouseover onmousemove onmouseout onkeypress onkeydown
onkeyup
```

يمكن تجنب حذف أو الخطأ في الكتابة في قائمة سمات عن طريق دمج كل السمات في مرجع معامل كينونة %Common.attrib وإذا أردت فيما بعد إضافة سمة إلى القائمة يمكن اضافتها عن طريق إضافتها على تعرييف Common.attrib ولا تجب اضافتة إلى تعريفات العناصر المطولة.

إتاحة تغير نماذج المحتوى وقوائم السمات لمختلف DTD

احدى الطرق البارعة في استخدام مراجع معاملات الكينونة في XHTML هو كيفية تخصيص ثلاثة DTD مختلفة من نفس أصل الواحدات النمطية الأساسية وأساسي تنفيذ ذلك هو أن كل علصر يمكن بعد ذلك لكل DTD أن تعيد تعريف نموذج المحتوى أو قائمة السمة بواسطة إعاده تعريف مرجع معامل الكينونة وهذا ينتج لبعض DTD المحددة إضافة أو حذف عناصر من نماذج المحتوى وقوائم السمات.

مثلاً في الوحدة النمطية XHTML1-table، ويتم تعريف عنصر caption كما يلي

```
<!ENTITY % Caption.content "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT caption %Caption.content; >
<!ATTLIST caption
    %Common.attrib;
>
```

لنفترض أن DTD يتطلب أن تحتوي العناوين على PCDATA غير لا تستخدم الترميز فمن السهل وضع تعريف الكينونة هذا في الملف الذي يستورد XHTML1-table.

```
<!ENTITY % Caption.content "( #PCDATA )" >
```

يتجاوز هذا التعريف في XHTML1-table حتى يمكن أن تتضمن العناوين التي تتصل DTD الخاص بك على نص ولا تحتوي على ترميز.

تحسين نماذج محتوى المستند

تستخدم مراجع معامل الكينونة في استبدال نوع سمة CDATA على الرغم من أن %ContentType; و %ContentTypes; و %Charset; و%Charsets; و %LanguageCode; و %Character; و %Number; و%LinkTypes; و %MediaDesc; و%URI على مستوى واحد وهي مرادفات CDATA فعلى مستوى أخر يجعلوا أنواع السمات اكثر تحديداً وقد تعني CDATA أي شيئ واستخدام كينونات المعاملات بهذه الصورة يؤدي إلى تضييق وتوثيق المعنى الفعلي في سياق محدد والتي لا تستطيع كينونات المعاملات هذه فرض معانيها وتوثيقها إلى اهداف بسيطة.

ضغط DTD عن طريق إعاده استخدام التتالي المعتاد للنص

تمثل XHTML DTD حوالي ٨٠ كيلو بايت وهي ليست كمية كبيرة خاصة بالنسبة لتطبيقات تعتمد على محرك أقراص محلي أو شبكة ولكن ليس شيئ هامشياً بالنسبة لتطبيقات الإنترنت فستكون غالباً ثلاثة إلى خمس اضعاف اكبر إذا تم توسيع قل مراجع معاملات الكينونات.

الأكثر تأثيراً حفظ حجم الملف الذي توصل إليه مراجع معاملات الكينونة هو الحفظ الواضح فالملفات الصغيرة يسهل قراءتها وفهمها فمثلاً مستند DTD حجمة ٦٠٠ كيلو بايت حتى وإن تم تقسيمة إلى اجزاء كل منها يحتوي على ٦٠ كيلو بايت سيكون مستند كبير لتطلب من الناشرين قراءتها واضعاً في الاعتبار التعليمات البرمجية التي تستخدم لغات بخلاف الانجليزية لصنع DTD أو لصياغتها في عبارة أخرى مثلما في هذا الفصل كم من التعليمات البرمجية قد تمت قراءتها فعلاً من البداية للنهايه وكم منها لم تتم قراءتها، فمن الطبيعي أن لا تقرأ أية تعليمات برمجية لوحدة نمطية تزيد عن الصفحة إلا إذا كنت قارئ جيد.

تقسيم DTD إلى مواحدات نمطية أحادية متصلة

يجعل تقسيم DTD إلى وحدات نمطية متصلة متعددة من السهل تجميعها تحلل ويتم تجميع كل مواد النذج في مكان واحد كما في مواد الجداول وكل مواد التطبيق وهكذا بالإضافة إلى ذلك يجعل هذا DTD اسهل في الفهم لأنه من الممكن التعامل من بأخذ قطعة يبلغ مجموعها بت واحد في المراة الواحدة.

من جهة أخرى يؤدى الترابط بين بعض الواحدات النمطية إلى القليل من التعارض اكثر مما يتطلب الأمر من أجل فهم أي من هذه الوحدات النمطية بصورة صحيحة يجب فهم -XHTML1 XHTML1-attribs.mod names.mod لأنها تقدم تعريفات دقيقة للكينونات المستخدمة في كل الوحدات النمطية الاخرى وبالاضافة إلى ذلك يمكن لوحدة نمطية أن تفهم في السياق الخاص باطار مجموعة DTD الثابت أو الواسع ولذا يجب فهم أربع ملفات قبل البدء في أي منها ومع ذلك فإن الانفصال بيين الوحدات النمطية تؤثر ويفرض نفسة في التقليد.

خلاصة

تعليمات في هذا الفصل:

- من المفترض أن يتعرق مؤلفي XML عمي الإصدارات السبقة للآخرين ا يقظة.
- يعتبر XHTML هو إصدار تم وضعة بلغة XML للغة HTML وياتي في ثلاثة طرق الثابت والواسع ومجموعة الأطر.
- يقوم XHTML DTD تقسم HTML إلى ٢٩ وحدة نمطية مختلفة وثلاث مجموعات كينونات.
- لا يمكن أن تحصل على الكثير من التعليقات في DTD مما يجعل الملف اسهل في القراءة.

- تعتبر كينونات المعاملات أدوات فعاله جداً لبناء DTD معقد ولكن يسهل التعامل والحفاظ علية.

في الفصل القادم سوف نكتشف تطبيق XML آخر وهو CDF الذي سيتم استخدامه لدفع المحتوى إلى المشتركين. وقمنا بالتركيز على XHTML DTD في هذا الفصل ولا يحتوي CDF على DTD منشور ولذلك سنتبع طريقة أخرى لنتمكن من فهمه.



الدفع بمواقع ويب عن طريق CDF

يعطي هذا الفصل تنسيق تعريف قناة Microsoft: "Channel Definition (CDF)"، والذي يعد تطبيق لغة XML لتعريف القنوات. تعتبر القناة مجموعة من صفحات ويب التي يمكن تقديمها إلى مشترك آلياً. يقوم مستند CDF بعمل قائمة للصفحات التي يتم تقديمها، والوسيلة التي تقدم بها والتردد إلي تقدم به، ومعلومات مشابهة. يستطيع القارئ أن يشترك في القنوات التي تستخدم Internet Explorer 4.0 وما بعده. بالإضافة إلى ذلك تستخدم صفحات ويب والقنوات لغة HTML الحيوية مثل JavaScript وJava لتنشئ تفاعل، يستمر في تحديث ساعات الأسهم، مربعات النتائج الرياضية ومثيلاتها. تستطيع القنوات المعروضة للقيود الأمنية أن تدفع بتحديث البرنامج إلى المستخدمين المسجلين وتثبتهم آلياً.

ما هو CDF

يعد تنسيق تعريف القناة (CDF) تطبيقات للغة XML متطورة على Microsoft لتعريف القنوات. تمكن القنوات. تمكن القنوات مواقع ويب من أخبار القارئ آليا بالتغيرات التي تمت في المعلومات الخطيرة يسمى أحياناً هذا الأسلوب Webcasting أو push وحالياً، يعد Internet Explorer هو المستعرض الكبير الوحيد الذي ينفذ CDF ويبدو الأكبر غير قابل للتنفيذ. لم يفعل W3C أي شئ اكثر من الاعتراف بمواصفات CDF رسمياً، ويبدو انهم لا يرغبون في أداء اكثر من ذلك في المستقبل.

يعتبر ملف CDF مستندXML ، منفصل عنة، ولكن مرتبط، بمستندات HTML في الموقع. يعرف مستند CDF المعاملات لإيجاد صلة بين القراء ومحتوى الموقع. من الممكن نقل البيانات من خلال push التي ترسل ملاحظاتها، أو حتى مواقع ويب بأكملها للقراء المسجلين أو من خلال قراء pull يختاروا تحميل الصفحة في مستعرض ويب الخاص بهم، ويحصلوا على المعلومات الحديثة.

لا تحتاج أن تعتمد كتابه موقعك لتحصل على مميزات CDF. يعد ملف CDF ببساطة إضافة للموقع. عامة رابطه ملف CDF على تحميل موقع صفحة رئيسية نسخة لفهرس القناة لجهاز القارئ. يضع هذا رمز على شريط قناة القارئ، التي يجوز أن تنقر للوصول إلى المحتويات الحالية للقناة.

كيف يتم إنشاء القنوات

لكي تنشئ قناة، أتبع تلك الخطوات الثلاثة

١ ـ قرر أي محتوى يجب أن تتضمنه القناة.

٢ ـ اكتب ملف تعريف القناة الذي يعرف المحتوى.

٣ـ اربط من الصفحة الرئيسية لموقع ويب إلى ملف تعريف القناة.

تحديد محتوى القناة

قبل أن تعجز عن التقدم في التفاصيل التقنية المتشابكة الكثيرة بشأن إنشاء قناة عن طريق CDF، يجب إن تقرر أولاً ما هي المحتويات الخاصة بالقناة وكيف يتم تسليمها.

يكون الاعتبار الأول عندما تحول المواقع الموجودة للقناة هو كم وأي صفحة من الصفحات تتضمنها. تقترح عوامل الواجهة البشرية أنها، لن تحتوي أي قناة على اكثر من ثمن بنود لكي يختار منهم القارئ. وألا سيصبح القراء في حيرة ويجدوا صعوبة في إيجاد ما يريدوا. بينما، يمكن ترتيب القنوات هيكلياً. من الممكن إضافة مستويات من المحتوى كقنوات فرعية. على سبيل المثال، قد يكون لقناة أخبار الصحف أقام للعمل، العلوم، التسلية، الأخبار العالمية، الأخبار القومية، والأخبار المحلية. يمكن تقسيم قسم التسلية إلى قنوات فرعية للتليفزيون، دور السينما، الكتب، الموسيقى، والرسم.

قد يتطابق المنظمة والهيكل المختارين أو لا مع منظمة أو هيكل موقع الويب الموجود الخاص بك، تماماً مثل عدم تطابق موقع ويب للهيكلة والمنظمة الخاص بك لهيكل ومنظمة للملفات على محرك أقراص ثابت للخادم بينما، سيجعل تطابق هيكل القناة لهيكل موقع ويب أسهل للحفاظ علية. يمكنك بالطبع تحديد صفحات معينة خارج الموقع وترتيبهم في هيكل محدد للقناة إذا بدت منطقية.

سيكون اعتبارك الثاني هو الطريقة التي تسلم بها المحتوى الجديد للمشتركين. عندما يتم الاشتراك في قناة، يتم منح القراء اختيار من بين ثلاث اختيارات.

١ـ يمكن إضافة القناة لشريط القناة ويجوز المشتركين أن يتأكدوا من ذلك كلما أرادوا.

٢ـ يمكن إخبار المشتركين بالمحتوى الجديد بواسطة البريد الإلكتروني وبعد ذلك يقوموا بتحميل القناة وقتما يريدوا.

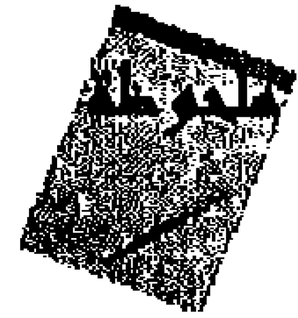
٣ـ يمكن فحص المستعرض للموقع دورياً لتحديث وتحميل المحتوي المتغير آلياً.

يمكن تصميم المحتوى الخاص بك ليعمل جيداً مع أياً من تلك الإخباريات الثلاثة التي يختارها القارئ.

إنشاء ملفات ومستندات CDF

بمجرد أن تقرر المحتوى الذي سيكون في قناتك، وكيف يتم تنظيم وتسليم ذلك المحتوى، فإنك تكون مستعد لكتابه مستند CDF الذي يقوم بتنفيذ تلك القرارات. يحتوي مستند CDF على

معلومات معرفة عن المحتويات الجدول والشعارات الخاصة بالقناة. يتم ترميز كل هذه المعلومات باستخدام مجموعة معينة من علامات XML يعد المستند المتبقي ملف XML المكون جيداً. يتم وضع تلك المستند على خادم الويب حيث يستطيع العملاء تحميلة.

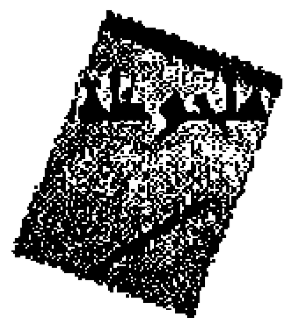


بينما سيكون غالباً أمر تافه أن تصمم DTD من أجل CDF وبينما أشك في أن Microsoft له واحد داخلياً، لم يقوموا بعد نشره لإصدار CDF الحالي من الممكن إيجاد DTD لإصدار CDF مبكر في ملاحظة W3C على http://www.w3.org/TR/NOTE-CDFsubmit.html بينما، لا تصبح هذه قريبة من وصف الإصدار الحالي CDF أن تكون مصممة جيداً، ولكن غير صحيحة.

يبدأ مستند CDF بتعريف XML لأن مستند CDF يعد مستند XML ويتبع نفس القواعد مثل جميع مستندات XML. يعد الجزر والعنصر المطلوب فقط لمستند channel CDF يجب لعنصر channel أن يكون له سمه HREF التي تحدد الصفحة التي تراقب للتغيرات. يعرف دائماً جزر عنصر channel الصفحة الرئيسية في القناة تعد تعليمات البرمجة ٣٢-٢ مستند CDF بسيط يشير إلى صفحة يتم تحديدها بالزيادة أو النقصان يومياً.

تعليمات البرمجة ٢١-١ : أبسط مستند CDF محتمل للصفحة

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
</CHANNEL>
```



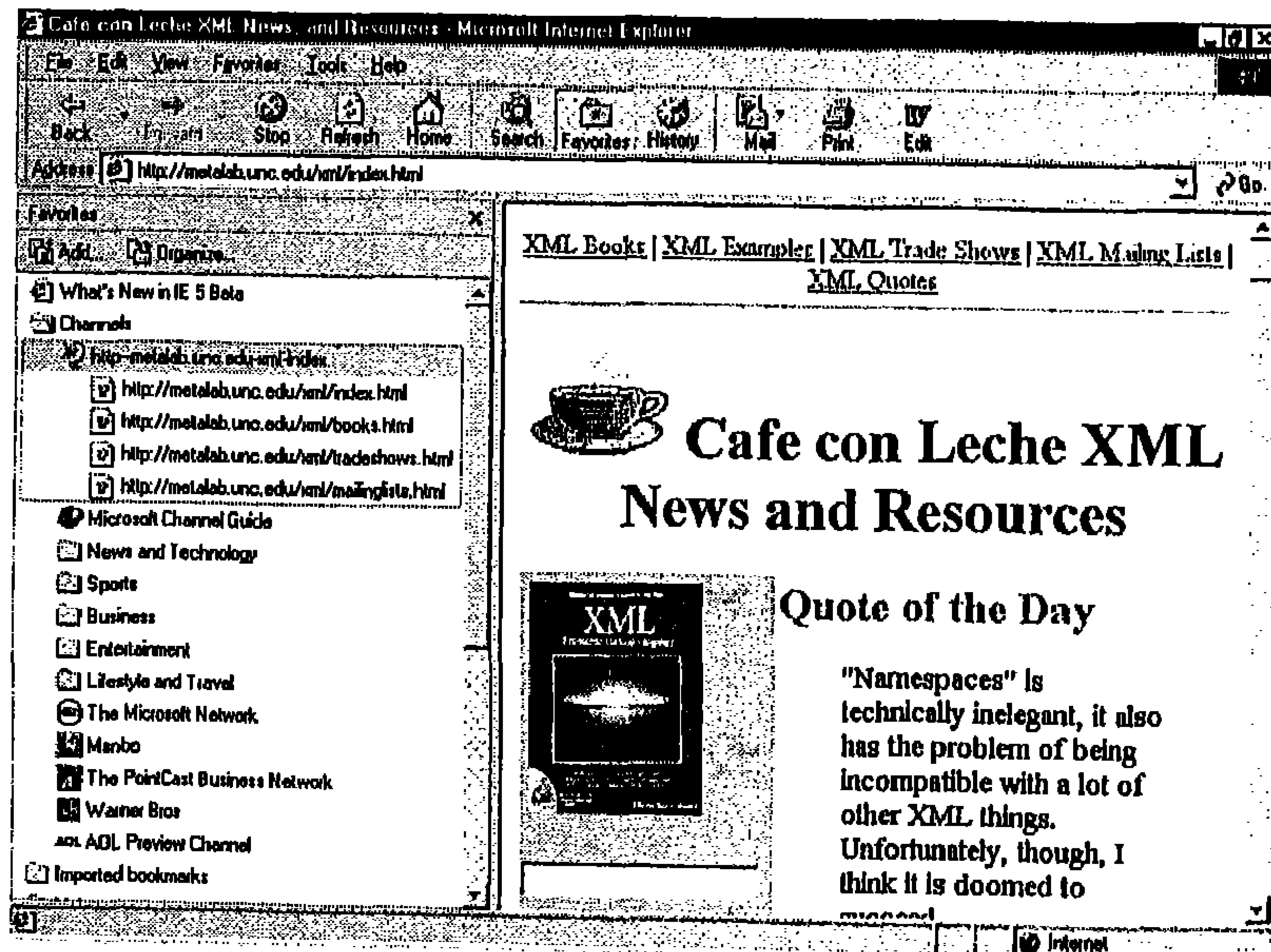
تعتبر معظم مستندات CDF Microsoft مستنده على إصدار سابق لموصفات XML التي تستخدم حالة الأحرف العليا <?XML version="1.0"?> بدلا من حالة الأحرف أنها تعمل مع Internet Explorer، لذلك ففي هذا الباب سأستخدم حالة الأحرف السفلى XML التي تطابق استخدام مقياس XML.

بالإضافة إلى الصفحة الرئيسية، تحتوي معظم القنوات على مجموعة صفحات أخرى تعرف بواسطة توابع ITEM. ولكل ITEM يوجد سمة HREF التي تشير إلى الصفحة. توضح تعليمات البرمجة ٣٢-٣ قناة تحتوي على صفحة رئيسية

(http://metalab.unc.edu/xml/index.html) مع ثلاث أفـــرع فرديـــة للصفـــح فـــي عناصر ITEM تظهر دائماً القنوات في نظره للمخطط التفصيلي القـــابل للطـــي الـــذي يســـمح للمستخدم بإظهار أو إخفاء البنود الفردية في القناة كما يختاروا. يوضح الشكل ٣٢-٢ توسع هـذه القناة في الشريط المفضل لـInternet Explorer

تعليمات البرمجة ٢١-٢: قناة CDF مع توابع ITEM

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html:">
  </ITEM>
</CHANNEL>
```



الشكل ٢١-١ مجلد القنوات المفتوح في Internet Explorer 5.0.

ربط صفحة ويب بالقناة

تجعل الخطوة الثالثة والأخيرة ملف CDF متوافر للقارئ ولكي تقوم بذلك، فإنك توفر رابطة من صفحة ويب إلى ملف CDF. تعد أسهل طريقة لتنفيذ ذلك مع مقياس عنصر A أن ينقر القارئ للتنشيط. عامة، ستكون محتويات هذا العنصر بعضها نصوص أو صور. تسال القارئ عن الاشتراك في القناة. على سبيل المثال.

```
<A HREF="cafeconleche.cdf">Subscribe to Cafe con Leche</A>
```

عندما يقوم القارئ بتنشيط هذه الرابطة في مستعرض ينتج التعامل مع CDF "وهو مجرد طريق خيالية لقول أو Internet Explorer 4.0 وما بعدها"، يحمل المستعرض ملف CDF المسمي في سمة HREF ويضيف القنوات إلى قوائم الاشتراك الخاص بها. أما المستعرضين الأخريين الذين لا يدعمون CDF من المحتمل أن يسألوا المستخدم أن يحفظ المستند كما هو واضح في الشكل ٣٢-٣.

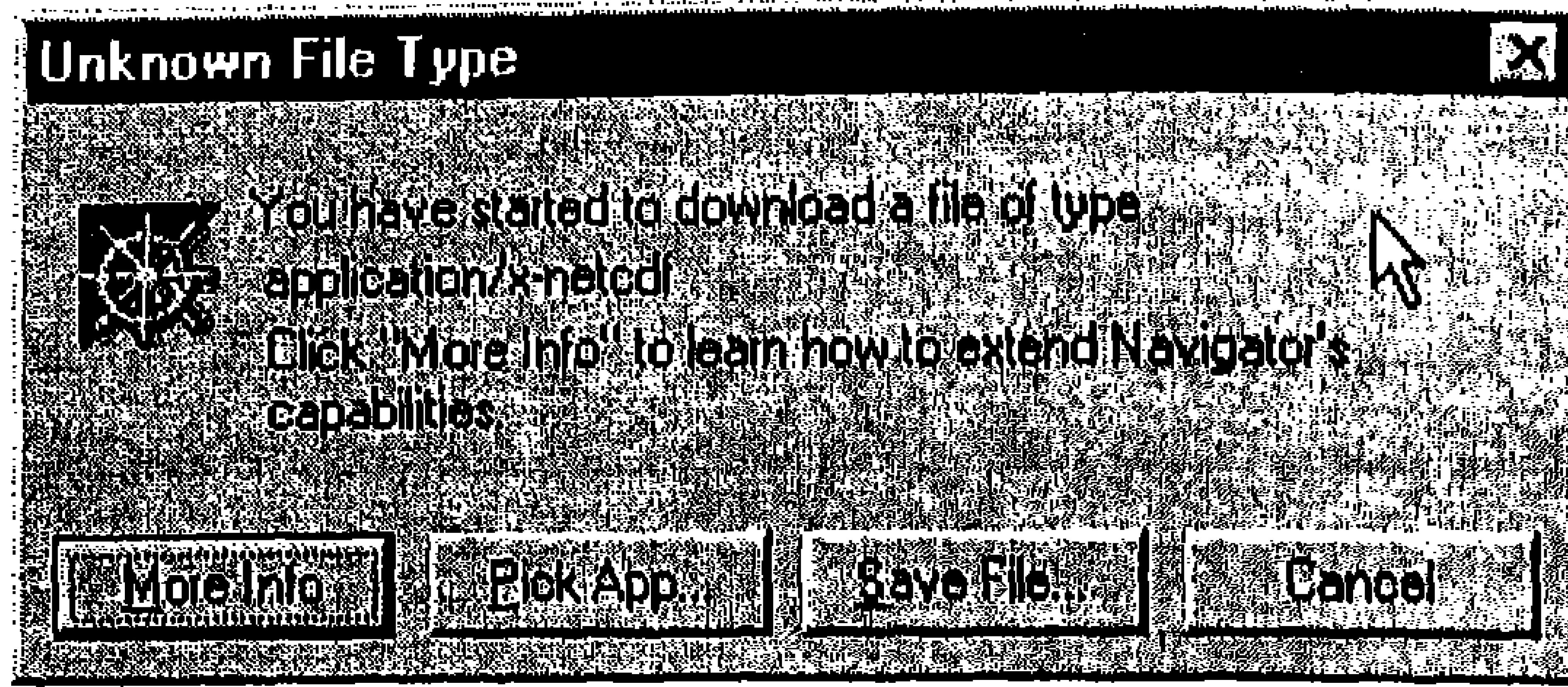
وبمجرد تحميل ملف CDF، سيسأل المستعرض المستخدم عن الطريقة التي يرغب إن يتم أخباره بها عن التغيرات المستقبلية للقناة كما هو واضح في الشكل ٣٢-٣. توجد ثلاث اختيارات للمستخدم:

١ـ يجوز إضافة القناة للمستعرض وأشرطة قناة الشاشة الرئيسية النشطة يجب أن يحدد المشترك يدوياً القناة ليحصل على التحديث. وهذا لا يختلف كلياً عن علامة الكتاب، إلا عندما يفتح المستخدم علامة للقناة يعاد تحديد جميع الصفحات التي في القناة بدلاً من تحديد صفحة واحدة فقط.

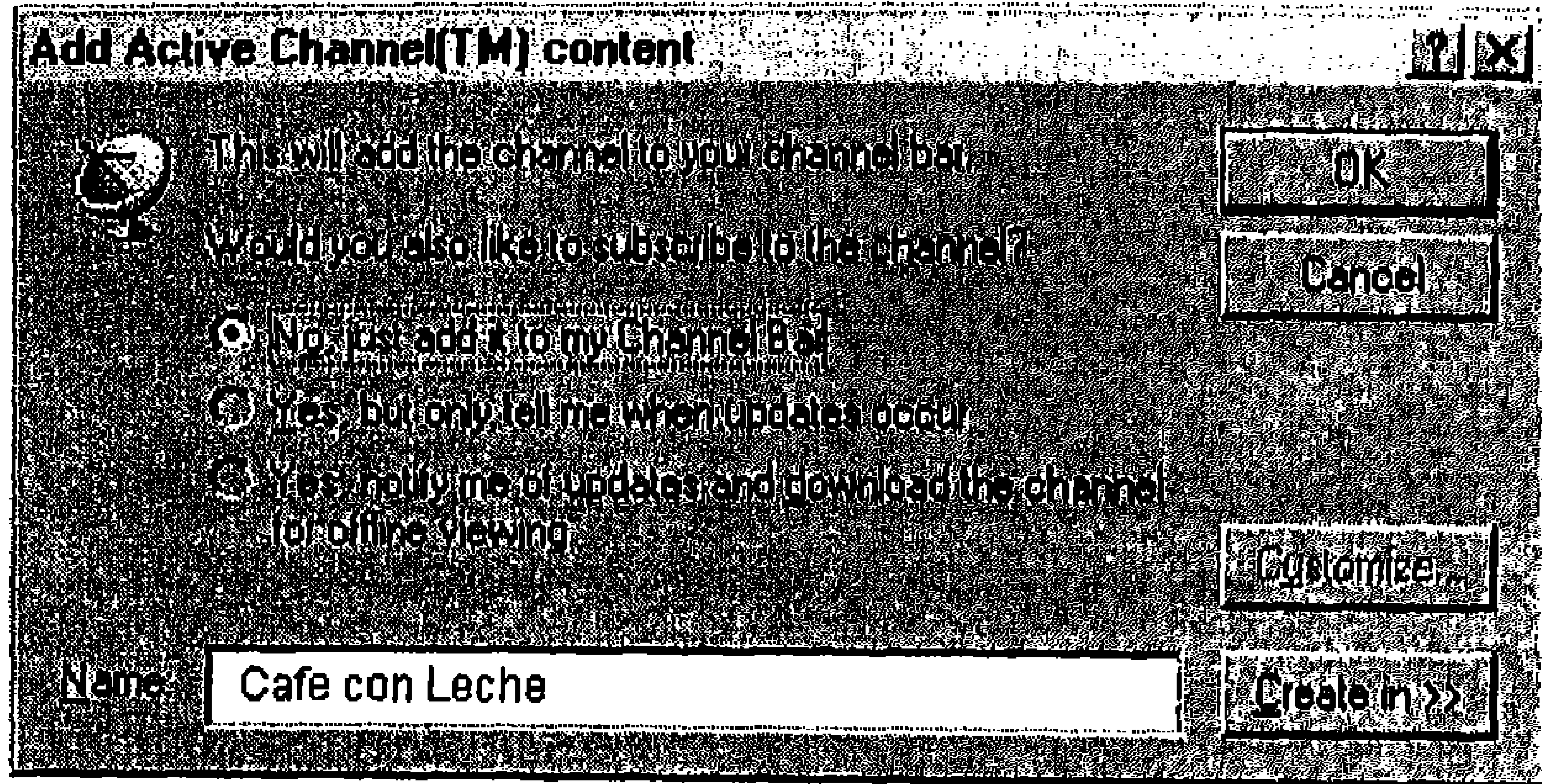
٢ـ يقوم المستعرض بفحص القناة دورياً لتحديث وإخبار المشترك بأي تغيرات بواسطة عن طريق البريد الإلكتروني يجب أن يستمر المستخدم في تحميل المحتوي الجديد.

٣ـ يقوم المستعرض بفحص القناة دوريا للتحديث وأخبار المشترك بأي تغيرات عن طريق البريد الإلكتروني. بينما عند ما يتم تحديد التغيير بحمل المستعرض ويقوم بتخزين مؤقت للمحتوى الجديد آليا لكي تصبح متوافرة على الفور للمستخدم كي يراها، حتى إذا لم يكونوا متصلين بالإنترنت أثناء فحصهم لموقع القناة.

تجعل تعليمات البرمجة ٣٢-٣ فقط الاختيار الأول متوافر لآن هذه القناة بالذات لا توافر جدول التحديث، ولكن سنضيف ذلك قريباً.



الشكل ٢١-٢ تنقل Navigator 5.0 لا يدعم CDF ولا يفـــهم ملف CDF.



الشكل ٢١-٣ يسأل Internet Explorer 4.0 المستخدم أن يختار الطريقة التي يرغب أن يخبروه بها عن التغيرات في الموقع.

وصف القناة

يجوز للقناة نفسها ولكل بند بالقناة أن يكون له عنوان، مجرد، وفيما يزيد عن ثلاثـــة شـــعارات بأحجام مختلفة يتـــم إنشـــاء هـــؤلاء بإعطـــاء عنـــاصر CHANNEL وITEM توابـــع TITLE وABSTRACT وLOGO.

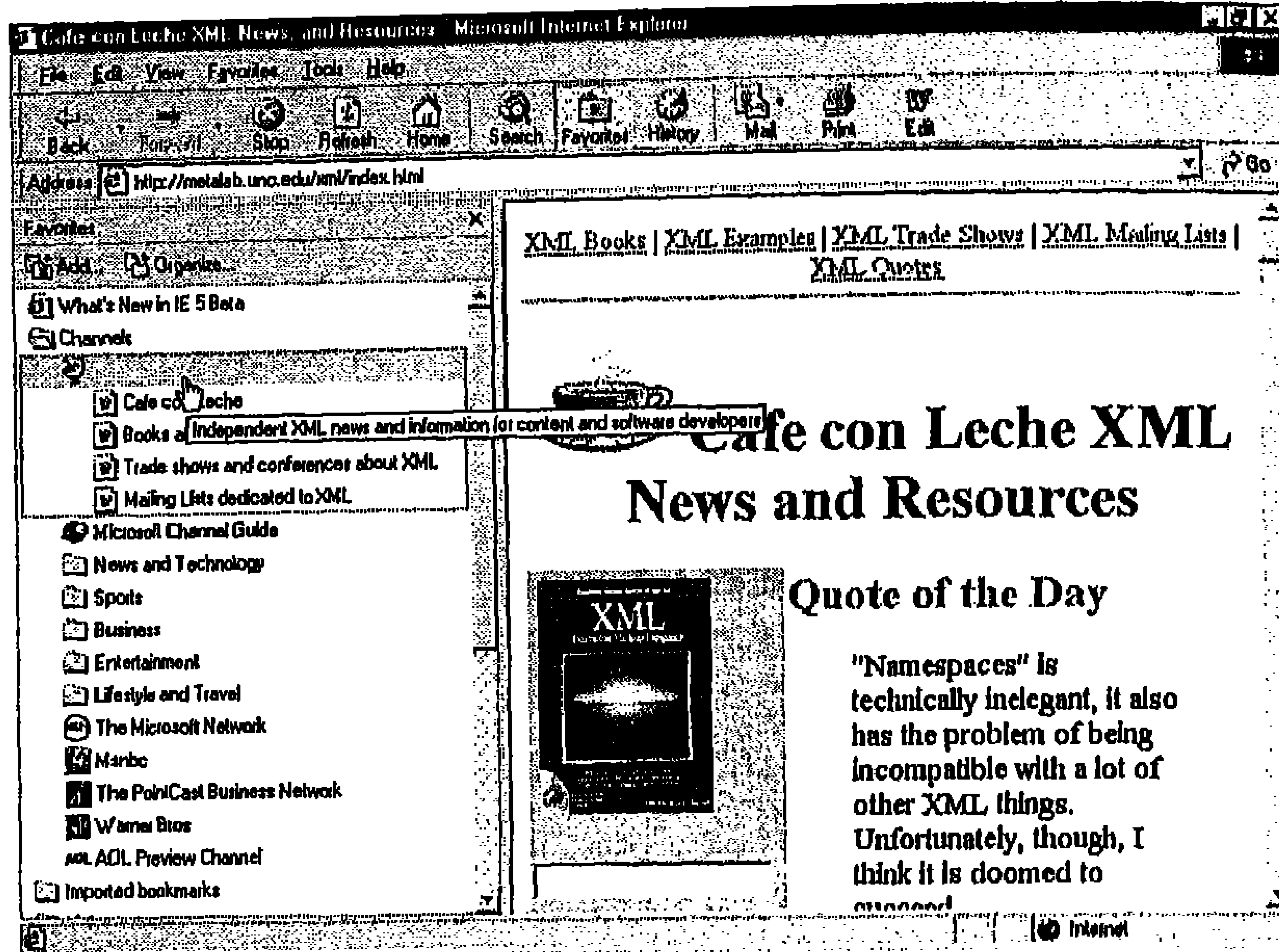
العنوان

لا يعد عنوان القناة مشابه تماماً لعنوان صفحة ويب. ولكن، يظهر عنوان القناة في هامش توثيـــق القناة، قائمة القناة، وشريط القناة، كما هو واضح في الشكل ٢١-٢ حيث يكون العنـــوان --http

metalab.unc.edu-xml-index على الرغم من أن المشترك له الخيار في تخصيصـــه عـن طريق كتابة عنوان مختلف كما هو واضح في الشكل ٣٢-٣. تستطيع أن توفر عنوان افـتراضي وصفي اكثر لكل عنصر CHANNEL وITEM بإعطائها تابع TITLE يحتـــوي كـل عنصـر TITLE فقط على بيانات حرفية، ولا يوجد أي ترميز. تصنيف تعليمات البرمجة ٣٢-٣ عنـاوين للصفح الفردية في قناة Cafe con Leche بالإضافة إلى القناة نفسها. يوضح الشكل ٣٢-٤ كيف يؤثر هذا على البنود الفردية في قائمة القناة.

تعليمات البرمجة ٢١-٣: قناة CDF بالعناوين

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <TITLE>Cafe con Leche</TITLE>
  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
    <TITLE>Books about XML</TITLE>
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
  </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
  </ITEM>
</CHANNEL>
```



الشكل ٢١-٤ تظهر العناوين في شريط القنوات وتظهر الأجـــزاء فـــي تعريف الأدوات.

التجربة

قد تكون العناوين كافية لقناه لها علامة تجارية جيدة مثل ديزني أو MSNBC، ولكن لبقيتنا توجـــد أضواء أقل في سماء الأخبار، فهي من المتحمل ألا تؤذي لتخبر المشتركين المزيد عما يتوقعـــوا أن يجدوه في موقع محدد. ولهذه النهاية، يجوز لكل عنصر CHANNEL وITEM أن يحتوي على عنصر فردى تابع ABSTRACT يجب أن يحتوي عنصر BSTRACT على كتلة قصـــيرة مـــن النص "٣١١ حرف أو أقل" تصف البند أو القناة عامة ستظهر تلك الصفات في ويندوز تعريـــف الأدوات كما هو واضح في الشكل ٣٢-٤، المستند على تعليمات البرمجة ٣٢-٤.

تعليمات البرمجة ٢١-٤: قناة CDF بالعناوين والملخصات

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
```

```
    </ABSTRACT>
  <ITEM HREF="http://metalab.unc.edu/xml/books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML
      with capsule reviews and ratings
    </ABSTRACT>
   </ITEM>

   <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
  </ABSTRACT>
   </ITEM>
  <ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
    <ABSTRACT>
      Mailing lists where you can discuss XML
    </ABSTRACT>
   </ITEM>
  </CHANNEL>
```

الشعارات

تستطيع مستندات CDF تحديد شعارات للقنوات. تظهر هذه الشعارات على جهاز القارئ إما على الشاشة الرئيسية أو في قائمة مستعرض القناة يجوز استخدام الشعارات في عدد من الطرق المختلفة داخل القناة: رموز على الشاشة الرئيسية، رموز في مشغل البرنامج شعارات في شريط القناة وكتيب إرشاد القناة. يجوز لكل عنصر CHANNEL وITEM ما يزيد عن ثلاث شعارات: واحد للشاشة الرئيسية واحد لمشغل البرامج، وواحد لشريط القناة.

يرتبط شعار محدد بقناة بها عنصر LOGO. يعد تلك العنصر تابع CHANNEL الذي يمثله . تعتبر سمة HREF لعنصر LOGO URL نسبي أو مطلق حيث يوجد ملف الرسومات الذي يحتوي على الشعار. يدعم Internet Explorer صور منشقة GIF وJPEG وICO للشعارات ولكن ليس GIFs متحركا. ولأن الشعارات قد تظهر ضد نطاق كامل من الألوان والنقوش على الشاشة الرئيسية، فإن GIFs بخلفيته الشفافة التي تقتصر على لوح ألوان ويندوز ذو النصف نعمة يعمل أفضل.

يعد لعنصر LOGO سمة STYLE الأزمة لتحديد حجم الصورة يجب أن تكون قيمـــة سـمة STYLE واحدة من الكلمات الأساسية الثلاثة ICON وIMAGE أو.IMAGE-WIDE يعدوا هـؤلاء أحجام مختلفة من الصور، كما هو موجود في الجدول ٣٢-٢ يوضح الشكل ٣٢-٥ الشـــعارات المستخدمة من أجل Cafe con Leche بالأحجام الثلاثة المختلفة.

الجدول ٢١-١

قيم لسمة STYLE الخاصة بعنصر LOGO

فيلد	حجلوكشفب ىكهشغ
ICON	يعرض رمز ١٦ بكسل عرض ١٦ بكسل ارتفاع في قائمة الملف وفـــي شريط القناة المجاور للعناصر التابعة في الهيكل، كما هو موضح فـــي الشكل ٢١-٢.
IMAGE	تعرض صورة الشاشة الرئيسية.
IMAGE-WIDE	تعرض صورة ١٩٤ بكسل عرض في ٣٢ بكسل ارتفاع في شريط قناة المستعرض. إذا تضمن هيكل القنوات في للأسفل، يظهروا عندما ينقـــر القارئ هذا الشعار، كما هو واضح في الشكل ٢١-٣.



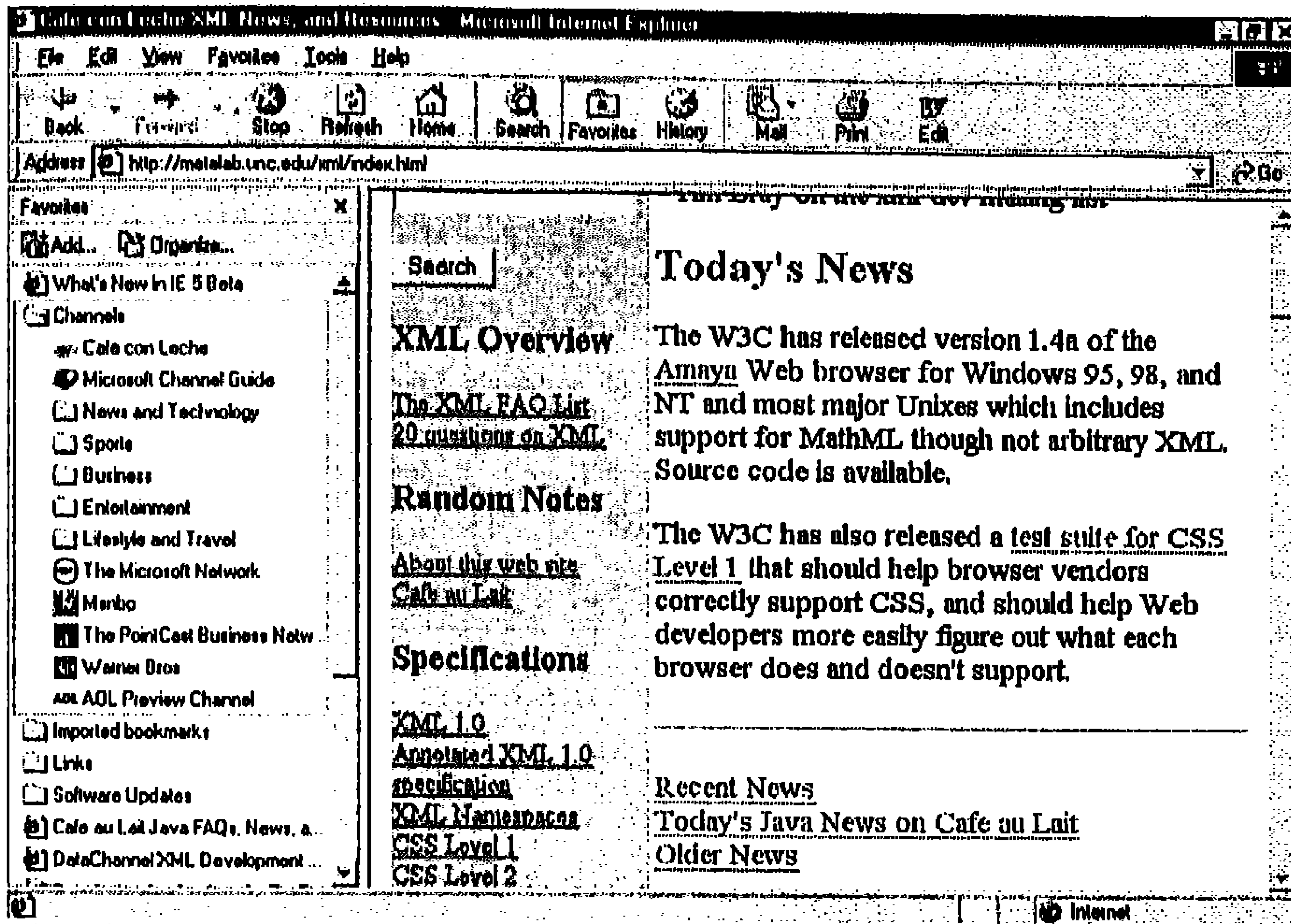


الشكل ٢١-٥ رموز قنـــاة The Cafe con Leche فـــي أحجامها الثلاثة المختلفة.

عندما يتغير المحتوى الموجود في القناة، يضع المستعرض ومضه مضيئة في الركن البـــاري الأعلى لصوره الشعار. تخفي تلك الوامضة أي شئ في هذا الركن. وأيضاً، إذا قام القارئ بعمـــل تمدد لعرض ويندوز يتعدي ٢٩٤ بكسل الموصي بهم، يستخدم المستعرض البكسل الذي يوجد في القمة اليمن لملئ الشعار المتوسع. وبالتالي تحتاج إلى أن تلفت انتباهك خاصة إلى الأركان أعلـــى اليمين واليسار للشعار.

معلومات لتحديث الجداول

تعد عناصر CHANNEL وTITLE وABSTRACT وLOGO كافين لبناء قنـــاة عمـــل، ولكنـــهم يوفروا فقط صلة مرئية قد يستطيع القراء أن يستخدموها في التوجه السريع إلى موقعك. بينما، لا يوجد لديك أي وسيلة ليقدم المحتوى إلى القراء القنوات غير الفعالة وهي، قنوات مثل الموجـــودة في تعليمات البرمجة ٣٢-٢ من خلال ٣٢-٥ التي ليس لديها جدول مقدم واضح لا يعملوا جيداً



الشكل ٢١-٦ تحتوي ألا شرطه المفضلة على رموز the Cafe con Leche بدلاً من رمز القناة العام.

تعد تعليمات البرمجة ٣٢-٥ مستند CDF الذي يوفر أحجام متنوعة من الشعارات. يوضـــح الشكل ٣٢-٦ الأشرطة المفضلة في Internet Explorer 5.0 وبها الشعار الجديد Cafe con Leche.

تعليمات البرمجة ٢١-٥ : قناة CDF بأحجام مختلفة من الشعارات

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">
  <TITLE>Cafe con Leche</TITLE>
```

```
<ABSTRACT>
  Independent XML news and information for content
  and software developers
</ABSTRACT>
<LOGO HREF="cup_ICON.gif"      STYLE="ICON"/>
<LOGO HREF="cup_IMAGE.gif"     STYLE="IMAGE"/>
<LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>
<ITEM HREF="http://metalab.unc.edu/xml/books.html">
  <TITLE>Books about XML</TITLE>
  <ABSTRACT>
    A comprehensive list of books about XML
    with capsule reviews and ratings
  </ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  <TITLE>Trade shows and conferences about XML</TITLE>
  <ABSTRACT>
    Upcoming conferences and shows with an XML focus
  </ABSTRACT>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
    Mailing lists where you can discuss XML
  </ABSTRACT>
</ITEM>
</CHANNEL>
```

لتقدم فعلياً المحتويات للمشترك، عليك بتضمين معلومات الجداول لتحديثها. يمكنك جدولة القناة بأكملها كجزء واحد أو الجدولة الفردية في القناة كلاً على حده. يتحقق هذا عن طريق إضافة العنصر التابع SCHEDULE إلى القناة. على سبيل المثال:

```
<SCHEDULE STARTDATE="29-03-1998" STOPDATE="29-.3-1998"
 TIMEZONE="0500-">
 <INTERVALTIME DAY="7"/>
 <EARLIESTTIME DAY="1" HOUR="0" MIN="0"/>
 <LATESTTIME DAY="2" HOUR="12" MIN="0"/>
</SCHEDULE>
```

يوجد لعنصر SCHEDULE ثلاث سمات STARTDATE وSTOPDATE وTIMEZONE STARTDATE تشير STARTDATE إلى الوقت الذي يبدأ فيه الجدول، وتشير STOPDATE إلى الوقت الذي ينتهي فيه الجدول. وهد فها هي الفترة التي تقع بين فحص الموقع المعتاد إذا غيرت بنية موقع ويب الخاص بك على فواصل زمنية منتظمة، استخدم هذا الفاصل الزمني نستخدم STARTDATE وSTOPDATE نفس تنسيق التاريخ: السنة العددية الكامل، شهر يتكون من رقمين، أيام الشهر المكتوبة من رقمين. على سبيل المثال ٣٢-٢٣-٢٩٩٩.

توضح سمة TIMEZONE الفرق في الساعات بين منطقة توقيت الخادم وتوقيت Greenwich Mean. إذا لم تضمن العلامة سمة TIMEZONE يظهر التحديث المجدول طبقاً لمنطقة توقيت القارئ وليس منطقة توقيت الخادم يعد التوقيت القياسي الشرق في الولايات المتحدة الأمريكية 0500-، ويعد التوقيت القياسي النصفي 0600-، والتوقيت القياسي للجبال 0700-، والتوقيت القياسي لمحيط الهادي 0800-، أما هاواي وألا سكا فهي 01000-.

يجوز أن يكون SCHEDULE لها ما بين عنصر واحد وثلاث عناصر تواريخ تعد INTERVALTIME عنصر مطلوب وفارغ يحدد كم مرة يجب على المستعرض أن يفحص القناة لتحديثها "فرضاً أن المستخدم قد سأل المستعرض أن يفعل ذلك". يكون لدى INTERVALTIME سمات DAY وHOUR وMIN. وتضاف سمات DAY وHOUR وMIN لحساب كمية الوقت المسموح بها لتنقضي بين التحديث. وبما إن هناك واحدة موجودة فإن الاثنين الآخرين يجوز حذفهما.

تعتبر عناصر EARLIESTTIME وLATESTTIME عناصر اختيارية تحدد الأوقات خلال فحص المستعرض للتحديث. يوزع تحميل الخادم المتبقي والذي تم تحديثه على الفاصل الزمني بين الأوقات المبكرة والمتأخرة. وإذا لم تحدد ذلك، يقوم المستعرض ببساطة بفحصه بطريقته. بعد EARLIESTTIME وLATESTTIME سمات DAY وHOUR المستخدمين لتحديد وقت حدوث التحدث. يتراوح DAY بين ٢ "الأحد" إلى ٧ "السبت". ويتراوح HOUR من صفر "منتصف الليل" إلى ٣٣ "٢٢: .مساءاً". على سبيل المثال، يوضح المثال الأعلى أن المستعرض يجب أن بتحديث القناة مرة كل أسبوع (=INTERVALTIME DAY) بين منتصف ليل الأحد (=EARLIESTTIME DAY= HOUR) ومساء الاثنين (=LATESTTIME DAY=HOUR).

يجوز أيضاً أن يكون EARLIESTTIME وLATESTTIME سمة TIMEZONE التي تحـــدد منطقة الوقت التي تحسب فيها الأوقات المبكرة والمتأخرة . إذا لم يتم تحديـــد منطقـــة الوقـــت، أستخدم منطقة وقت القارئ لتحديد الأوقات المبكرة والمتأخرة. ولنفترض التحديث على منطقـــة وقت محددة، اعمل على تضمين السمة الإخبارية TIMEZONE في علامـات EARLIESTTIME وLATESTTIME. على سبيل المثال:

```
<EARLIESTTIME DAY="1" HOUR="0" TIMEZONE="0500-" />
<LATESTTIME DAY="2" HOUR="12" TIMEZONE="0500-"/>
```

لكي تقدم تحديث عبر LAN، يمكنك اختيار يوم الأسبوع "على سبيل المثال الأحد" والامتـــداد الزمني "من منتصف الليل للخامسة صباحاً". يقوم جميع المستعرضين بالتحديث أثناء فترة تكـــون خمس ساعات إذا قمت بالتحديث عبر صلات الإنترنت، يجب اتصال القراء بـــالإنترنت ليقـــوم المستعرض بتحديث القناة.

تتوسع تعليمات البرمجة ٣٢-٦ قناة Cafe con Leche لتتضمن التحديث المجدول. وبما أنه يتم تحديث المحتوى غالباً يوماً يتم تعين INTERVALTIME في يوم واحد. معظم الأيام التي يتـــم فيه التحديث تكون بين الساعة ٧ صباحاً والساعة ٢٣:-١ مساءاً بالتوقيت الشـــرقي. وبالتـــالي، تعين EARLIESTTIME ٢١ صباحاً EST وتعين LATESTTIME ٢٣ مساءاً EST لا يوجـد أي تاريخ بداية أو نهاية محدد من أجل التغيرات التي تحدث لهذا المحتـــوي، لذلـــك فـــإن ســـمات STARTDATE وSTOPDATE يقم حذفهما من الجدول.

تعليمات البرمجة ٢١-٦: قناة CDF بتحديث مجدول

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">

  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
  <LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
  <LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
  <LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>

  <SCHEDULE TIMEZONE="0500-">
    <INTERVALTIME DAY="1"/>
```

```
  <EARLIESTTIME HOUR="10" TIMEZONE="0500-"/>
  <LATESTTIME HOUR="12" TIMEZONE="0500-"/>
 </SCHEDULE>

 <ITEM HREF="http://metalab.unc.edu/xml/books.html">
  <TITLE>Books about XML</TITLE>
  <ABSTRACT>
   A comprehensive list of books about XML
   with capsule reviews and ratings
  </ABSTRACT>
 </ITEM>

 <ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  <TITLE>Trade shows and conferences about XML</TITLE>
  <ABSTRACT>
   Upcoming conferences and shows with an XML focus
  </ABSTRACT>
 </ITEM>

 <ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
   Mailing lists where you can discuss XML
  </ABSTRACT>
 </ITEM>
</CHANNEL>
```

التخزين المؤقت السابق والتقدم البطيء للويب

إذا اختار المشترك إن يحمل محتويات القناة آلياً عندما يتغير ولا وبعد ذلك يكون لصاحب الموقع الخيار في السماح للمشتركين برؤية الصفحات غير المتصلة وأيضا لتحميل اكثر من مجرد تلك الصفحات المعرفة في مستند CDF. يمكنك أن تسمح للمستعرض بأن ينتشر خلال موقعك، ويحمل صفحات إضافية ما بين المستوي الأول والمستويات الثلاثة على عمق من الصفحات المحددة.

التخزين المؤقت السابق

افتراضياً، يقوم المستعرضين بالتخزين المؤقت المسبق للصفحات الموجودة على القائمة في القنـاة للمستعرضات غير المتصلة، إذا طالب المستخدم أن يقوموا بذلك. بينما، يمكن أن يمنـع الكـاتب صفحة من أن يتم تضمين سمة PRECACHE في عنصر CHANNEL or ITEM بقية NO علـى سبيل المثال:

```
<CHANNEL PRECACHE="NO"
     HREF="http://metalab.unc.edu/xml/index.html">
</CHANNEL>
```

إذا كانت قيمة PRECACHE هي NO، أذان لن يتم التخزين الموقت السابق للمحتوى بغـض النظر عن إعدادات المستخدم. أما كانت قيمة PRECACHE هي YES "أو لا يكون هنـاك سـمة PRECACHE الواضحة" ويطالب المستخدم بالتخزين المؤقت السابق عند اشـتراكهم، إذن سـيتم تحميل المحتوى آليا بينما، إذا لم يطالب المستخدم بالتخزين المؤقت السابق، إذن فإن قناة الموقـع لن يتم تخزينها المؤقت السابق بعض النظر عن قيمة سمة PRECACHE.

عندما تقوم بتصميم قناة، يجب أن تأخذ في اعتبارك أن بعض القراء سيشاهدون المحتوى غير المتصل غالباً واضحاً وكنتيجة لذلك، ستكون أي روابط في محتويات القناة منتهيـة المفعـول إذا كنت تقدم المستندات عبر الإنترنت لن يكون لاختيار للتخزيـن المؤقـت معفـي كبـير، لأنـك ستضاعف بهذا نفس الملفات على الأقراص عبر الدمج. إذا قمت بتسليم المحتوى للقـراء الذيـن يدفعون للوقت الفوري، قد تريد أن تنظمها حتى يتم تخزينها مؤقتاً وتكـون مسـتعرض متصـل بسهولة. بينما لا تعد فكرة التقدم البطيء لويب بهذا العمق فكرة جيدة. لا تتغير معظم الصفحـات على الموقع يومياً وسيتم فحصهم في كل تحديث.

تقدم الويب ببطء

لا يكون المستعرضين محددين فقط بتحميل صفحات ويب المتخصصة في عناصر CHANNEL وITEM إذا كان لعنصر CHANNEL وITEM سمة LEVEL بقيمة أعلى من صفـر، فسـيقوم المستعرض بتقدم ويب ببطيء ثناء التحديث. يجعل تقدم ويب البطيء المستعرض يجمع صفحات اكثر من الصفحات الموجودة بالقائمة في القناة. على سبيل المثال، إذا كانت الصفحة الموجـودة بالقائمة في القناة تحتوي على عدد من الروابط المتصلة بالموضوعات الرئيسية، فمن السهل جعل المستعرض يحملها كلها بدلاً من إدراجهم في القائمة في عنـاصر ITEM الفرديـة. وإذا كـان للموقع هيكلية مضبوطة. يمكنك بأمان إضافة سمة LEVEL لعلامة القناة القصوى بالقمة، ويمكـن السماح لتقدم ويب البطيء في أن يتضمن جميع الصفحات بمستويات متتالية. يتراوح LEVEL بين الصفر "الافتراضي" ورقم ثلاثة، مما يشير إلى أي مدي بعيد للأسفل في هيكله الرابطة تريـد أن

ينقب عنها المستعرض أثناء التخزين المؤقت للمحتوى. تعد الهيكلية هيكلية تجريدية تعرف عن طريق روابط المستند، ولا تعرف الهيكلية عن طريق البنية المباشرة للملفات على خادم ويب تعتبر الصفحات التي لها إطار على نفس مستوى الصفحة التي بها مجموعة أطر، على الرغم من الاحتياج إلى رابطة إضافية من أجل المنفذ ولسمة LEVEL معنى حقيقي فقط إذا تمكنت من التخزين الموقت السابق.

تعين تعليمات البرمجة ٣٢-٧ LEVEL الخاص بقناة Café Con Leche إلى ثلاثة. يعمل هذا بعمق كافي ليصل إلى كل صفحه على الموقع. بما أنة تمت إشارة الصفحات مسبقاً إلى توابع ITEM، بأنها مستوى واحد فقط لأسفل من الصفحة الرئيسية، لا تكون هناك الحاجة الماسة لإدراجهم في القائمة على حدة. يمكن لعنصر LOGTARGET أن يكون له تابع PURGETIME الذي له سمة HOUR التي تحدد عدد الساعات التي تعتبر فيها معلومات تسجيل الدخول صحيحة. ويجوز أيضا أن يكون لها أي عدد من توابع HTTP-EQUIV المستخدمة لتعين أزواج من القيمة الأساسية المحددة في لرأس HTTP MIME. توضح تعليمات البرمجة٢١-٨ قناة تسجيل دخول لها صله بالقارئ.

تعليمات البرمجة ٢١-٧: قناة CDF التي تخزن مؤقتاً وسابقاً الثلاث المستويات بعمق

```
<?xml version="1.0"?>
<CHANNEL LEVEL="3"
      HREF="http://metalab.unc.edu/xml/index.html">
<TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
   Independent XML news and information for content
   and software developers
  </ABSTRACT>
  <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
  <LOGO HREF="corp_logo_regular.gif" STYLE="IMAGE"/>
  <LOGO HREF="corp_logo_wide.gif" STYLE="IMAGE-WIDE"/>
<SCHEDULE TIMEZONE="0500-">
   <INTERVALTIME DAY="1"/>
   <EARLIESTTIME HOUR="10" TIMEZONE="0500-"/>
   <LATESTTIME HOUR="12" TIMEZONE="0500-"/>
  </SCHEDULE>
</CHANNEL>
```

تسجيل دخول القارئ

يعد أحد عيوب القنوات مقارنة بمظهر ويب التقليدي هو عدم معرفة الخادم أي صفحة يراها بالفعل المستخدم. قد يكون هذا مهماً تتبع الإعلانات من بين إنشاء أخرى يستطيع Internet Explorer مراقبة مرور القارئ خلال موقع مخزن مؤقت غير متصل، وبعيدة ثانية إلى خادم ويب. بينما، يكون دائما للمستخدم الخيار في تعطيل هذا التصرف إذا شعر أنه انتهاك للخصوصية.

ولتجميع الإحصائيات عن الاستعراض متصل للموقع، تضيف عناصر LOGو LOGTARGETإلى عنصر CHANNEL أثناء تحديث القناة، يرسل الخادم محتويات القناة الجديدة للمستعرض بينما يرسل المستعرض ملف تسجيل دخول إلى الخادم يكون دائماً لعنصر LOG هذا الشكل، بالرغم من إن القيم الممكنة الأخرى لسمة VALUE يمكن إضافتها في المستقبل.

```
<LOG VALUE=document:view/>
```

ولعنصر LOGTARGET سمة HREF التي تعرف URL وسيتم إرسالها سمة METHOD التي تعرف طريقة HTTP مثل PUT أو POST التي ستستخدم لتحميل الملف تسجيل الدخول، وسمة SCOPE التي لها واحد من القيم الثلاثة! ALL, ONLINE, or OFFLINE التي تشير إلى أي مظهر للصفحة سيتم حسابه.

تعليمات البرمجة ٢١-٨: قناة CDF بتقرير تسجيل دخول

```
<?xml version="1.0"?>
<CHANNEL HREF="http://metalab.unc.edu/xml/index.html">

  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
  <LOGO HREF="logo_icon.gif" STYLE="ICON"/>
  <LOGO HREF="corp_logo_regular.gif" STYLE="IMAGE"/>
  <LOGO HREF="corp_logo_wide.gif" STYLE="IMAGE-WIDE"/>

  <LOG VALUE="document:view"/>
  <LOGTARGET METHOD="POST" SCOPE="ALL"
   HREF="http://metalab.unc.edu/xml/cgi-bin/getstats.pl" >
```

```
  <PURGETIME HOUR="12"/>
  <HTTP-EQUIV NAME="ENCODING-TYPE" VALUE="text"/>
</LOGTARGET>

<SCHEDULE TIMEZONE="0500-">
  <INTERVALTIME DAY="1"/>
  <EARLIESTTIME HOUR="10" TIMEZONE="0500-"/>
  <LATESTTIME HOUR="12" TIMEZONE="0500-"/>
</SCHEDULE>

<ITEM HREF="http://metalab.unc.edu/xml/books.html">
  <TITLE>Books about XML</TITLE>
  <ABSTRACT>
    A comprehensive list of books about XML
    with capsule reviews and ratings
  </ABSTRACT>
  <LOG VALUE="document:view"/>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/tradeshows.html">
  <TITLE>Trade shows and conferences about XML</TITLE>
  <ABSTRACT>
    Upcoming conferences and shows with an XML focus
  </ABSTRACT>
  <LOG VALUE="document:view"/>
</ITEM>

<ITEM HREF="http://metalab.unc.edu/xml/mailinglists.html">
  <TITLE>Mailing Lists dedicated to XML</TITLE>
  <ABSTRACT>
    Mailing lists where you can discuss XML
  </ABSTRACT>
</ITEM>
</CHANNEL>
```

يتم كتابة العناصر فقط التي لها توابع LOG في ملف تسجيل دخول على سبيل المثال، يصل إلى:

http://metalab.unc.edu/xml/index.html,

http://metalab.unc.edu/xml/books.html و http://metalab.unc.edu/xml/ tradeshows.html سيصبح تسجيل الدخول بينما يصل إلــى http://metalab.unc.edu/ xml/mailinglists.html لن يكون تسجيل دخول.

تخزن المعلومات الخاصة بتسجيل دخولCDF في التنسيق تســجيل دخــول ملحــق الملــف المستخدم بواسطة أغلب خادم ويب الحديثة بينما، يجب ترتيب خادم الويب، عامــة مــن خــلال برنامج في الخادم الرئيسي لتسجيل الدخول.

يجب أن يظهر عنصر LOGTARGET كتابع لعلامة CHANNEL بمستوى القمــة، وتصــف ملف تسجيل الدخول لجميع البنود التي تحتويها. بينما، يجب أن يتضمن كل عنصر ملف تســجيل الدخول لجميع البنود التي تحتويها. بينما،يجب أن يتضمن كل عنصر CHANNEL وITEM التــي تريد تضمينه داخل تسجيل الدخول، تابع LOG الخاص به.

سمة BASE

لقد استخدمت جميع الأمثلة السابقة URLs المطلقة لعناصر CHANNEL وITEM. بينمــا، تعــد URLs المطلقة غير مقنعة. لأنهم دائماً أطوال وأسهل في كتابتهم خطأ، وهذا سبب. أما الســبب الأخر، فلأنهم يجعلوا صيانة الموقع صعبه عندما تتحرك الصفحات من جهة إلى أخرى، أو مــن موقع إلى أخر، تستطيع استخدام URLs نسبية بدلاً من ذلك إذا عملت على تضمين ســمة BASE في عنصر CHANNEL.

تعد قيمة سمة BASE URLs التي تكون URLs النسبية في القناة نسبية لها على سبيل المثــال، إذا تعين BASE إلى /http://metalab.unc.edu/ xml إذن فإن سمة HREF ستكون ببســاطة books.html بدلاً من /http:// metalab.unc.edu/xml توضح ذلك تعليمــات البرمجيــة ٢١-٩.

تعليمات البرمجة ٢١-٩: قناة CDF بسمه BASE

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
   Independent XML news and information for content
   and software developers
```

```
  </ABSTRACT>
  <LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
  <LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
  <LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>
  <ITEM HREF="books.html">
    <TITLE>Books about XML</TITLE>
    <ABSTRACT>
      A comprehensive list of books about XML

with capsule reviews and ratings
      </ABSTRACT>
  </ITEM>
  <ITEM HREF="tradeshows.html">
    <TITLE>Trade shows and conferences about XML</TITLE>
    <ABSTRACT>
      Upcoming conferences and shows with an XML focus
      </ABSTRACT>
  </ITEM>

  <ITEM HREF="mailinglists.html">
    <TITLE>Mailing Lists dedicated to XML</TITLE>
    <ABSTRACT>
      Mailing lists where you can discuss XML
      </ABSTRACT>
  </ITEM>
</CHANNEL>
```

يمكنك لأي مكان نستخدمه لربط المحتوى، استخدام URL يسمي فــي العنــاصر التابعــة إذا حددت سمة BASE في عنصر CHANNEL الأصلي. تغير أيضا سمة BASE الهيكل المعــروض في Internet Explorer. ستعرض الصفحة المستندة في مستعرض ويندوز عندما لا يتم ربــط العناصر التابعة بالصفحة.

سمه LASTMOD "التعديل الأخير"

عندما يقوم المستعرض بعمل طلب لخادم الويب يرسل الخادم رأس MIME مع الملف المطلــوب تتضمن تلك الرأس أجزاء مختلفة من المعلومات مثل نوع الملف MIME، طول الملف، التــاريخ والوقت الحاليين، والوقت الذي تم فيه التعديل الأخير للملف. على سبيل المثال.

```
HTTP/1.1 200 OK
Date: Wed, 27 Jun 1999 21:42:31 GMT
Server: Stronghold/2.4.1 Apache/1.3.3 C2NetEU/2409 (Unix)
Last-Modified: Tue, 20 Oct 1998 13:15:36 GMT
ETag: "4b94d-c70-362c8cf8"
Accept-Ranges: bytes
Content-Length: 3184
Connection: close
Content-Type: text/html
```

إذا أرسل المستعرض طلب HEAD بدلاً من طلب GET الأكثر شيوعاً، يتم فقط إعادة الرأس. وعندئذ يجوز للمستعرض أن يفتش عن رأس التعديل الأخير لتحديد إذا كان الملف السابق تحميلة من القناة يحتاج لا عادة تحميله أم لا بينما، على الرغم من أن طلبات HEAD تعد أســـرع مـــن طلبات GET، مازال الكثير منهم يقوموا باستهلاك موارد الخادم.

لكي تقطع تحميل تحديث القناة المتتالية الموضوعة على الخادم الخاص بك، يمكنك أن تضيف سمة LASTMOD لجميع علامات CHANNEL وITEM. وعلى المستعرض أن يقوم فقط بإعــادة الفحص مع الخادم الأوقات المعدلة لتلك البنود والقنوات التي لا توفر سمات LASTMOD.

تعد قيمة سمة LASTMOD تاريخ ووقت، مثل 2000-05-23T21:42 عندما كان أخر شــئ تغير هو الصفحة المشار أليها بسمة HREF. يحـــدد ويقـــارن المســـتعرض التـــاريخ المعطـــي LASTMOD في ملف CDF بتاريخ المعدل الأخير المزودة بخادم ويب. عندما تغير محتوى خادم ويب، يتم تحديث التخزين المؤقت مع المحتوى الحالي. بهذه الطريقة يحتاج المستعرض فقط إلــى فحص ملف واحد، مستند CDF، للأوقات المعدلة بدلاً من كل ملف والذي يعد جزء مـــن القنـــاة. توضح تعليمات البرمجة ٢١-٣٢ ذلك

تعليمات البرمجة ٢١-١٠: قناة CDF مع سمات LASTMOD

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/"
```

```
      LASTMOD="27-01-1999T12:16" >
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
   Independent XML news and information for content
   and software developers
  </ABSTRACT>
  <LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
  <LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
  <LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>
  <ITEM HREF="books.html" LASTMOD="03-01-1999T16:25">
   <TITLE>Books about XML</TITLE>
   <ABSTRACT>
    A comprehensive list of books about XML
    with capsule reviews and ratings
   </ABSTRACT>
  </ITEM>

  <ITEM HREF="tradeshows.html" LASTMOD="10-01-1999T11:40">
   <TITLE>Trade shows and conferences about XML</TITLE>
   <ABSTRACT>
    Upcoming conferences and shows with an XML focus

 </ABSTRACT>
  </ITEM>

  <ITEM HREF="mailinglists.html" LASTMOD="06-01-1999T10:50">
   <TITLE>Mailing Lists dedicated to XML</TITLE>
   <ABSTRACT>
    Mailing lists where you can discuss XML
   </ABSTRACT>
  </ITEM>
 </CHANNEL>
```

عملياً، تعد هذه الطريقة من الصعب القيام بها يدوياً، خاصة بالشبه للمستندات المتغيرة باستمرار "والنقطة الكاملة للقنوات والتقدم هي أنهم يوفروا المعلومات التي تتغير باستمرار". بينما، يمكنك كتابة مستند CDF كملف ملئ بجانب الخادم الذي يتضمن ذلك ويدمج آلياً قيم LASTMOD في التنسيق المناسب أو يبتكر بعض الحلول البرمجية الأخرى بدلاً من التعديل اليدوي لسمه LASTMOD كل مرة تحرر فيها ملف.

عنصر USAGE

يحتوي عنصر CHANNEL و ITEM على العنصر التابع USAGE الاختياري الذي يعمل على حجم أحرف المشترك يحدد معنى عنصر USAGE بواسطة سمة VALUE الخاصة به. وتعد القيم الممكنة لسمة VALUE:

- ♦ Channel
- ♦ DesktopComponent
- ♦ Email
- ♦ NONE
- ♦ ScreenSaver
- ♦ SoftwareUpdate

يعد معظم وقت USAGE عنصر فارغ. على سبيل المثال:

```
<USAGE VALUE="screenSaver" />
```

تعد القيمة الافتراضية USAGE هي Channel. البنود التي تستعمل القناة تظهر في شريط القناة المستعرض تكون لكل عناصر CHANNEL و ITEM الذين رأيتهم حتى الآن استعمال CHANNE، على الرغم من أنهم ليس لديهم عنصر استعمال واضح. تسمح القيم الأخرى USAGE لمختلف المستخدمين بمواجهه محتوى القناة.

قيمة مكون الشاشة الرئيسة

تعد مكونات الشاشة الرئيسية صفات أو صور ويب صغيره تعرض مباشرة على الشاشة الرئيسية الخاصة باستخدام. وبما أن صفحاً ويب يمكن إن تحتوي على تطبيق Java الصغير، DHTML الخيالي، أو عنصر تحكم ActiveX، يمكن لمكون الشاشة الرئيسية أن يكون بالفعل برنامج "فرضاً أن المشترك قد ترك جميع أشكال التحذير والشاشة الرئيسية النشطة المثبتة".

يتم تثبيت مكون الشاشة الرئيسية على الشاشة الرئيسية الخاصة بالمشترك بمستند CDF منفصل يحتوي على عنصر ITEM الذي يشير إلى المستند المقرر عرضه على شاشة المستخدم الرئيسة. وبالإضافة إلى العناصر التابعة المعتادة، يجب أن يحتوي ITEM على عنصر USAGE غير الفارغ الذي يكون VALUE الخاصة به هو الشاشة الرئيسية. قد يحتوي عنصر USAGE على توابع OPENAS و HEIGHT و WIDTH و CANRESIZE.

تحدد سمة VALUE الخاصة بعنصر OPENAS نوع الملف في المكانة في سمة HREF لعنصر ITEM. قد يكون هذا إما HTML أو Image. إذا لم يكون موجود أي عنصر OPENAS، تفترض Internet Explorer أنه ملف HTML.

تحديد سمات VALUE لعناصر HEIGHT و WIDTH، عدد بكسل الذي يشغله البنــــد علــى الشاشة الرئيسية.

تشير سمة VALUE لعنصر CANRESIZE إلى كون القارئ قادر على تغير ارتفاع وعــرض المكون سريعاً. تعد قيمة الممكنة هي Yes أو No تعد Yes افتراضية. يمكنك أيضا السماح إعــادة تحجيم رأسياً أو أفقياً مستقلة عناصر CANRESIZEX،CANRESIZEY.

تعد تعليمات البرمجة ٣٢-٢٢ مكون بسيط للشاشة الرئيسية يستعرض صوره حقيقي للشــمس كما هي متوافرة بواسطة العاملين في المرصد القومي للطاقة الشمسية في Sunspot، بالمكســيك الجديدة (http://vtt.sunspot.noao.edu/gifs/video/sunnow.jpg) يعد ارتفاع الصــورة ٦٤١ بكسل، وعرضها ٤٨١ بكسل، ولكن يمكن ضبط حجمها. تحديث الصورة كل دقيقــة بيــن ٦:١١ صباحه MST و ٧:١١ مساءاً MST "لا توجد أي نقطة تجديد الصورة في الليل".

تعليمات البرمجة ٢١-١١: قناة مكون الشاشة الرئيسية

```
<?xml version="1.0"?>
<CHANNEL HREF="http://vtt.sunspot.noao.edu/sunpic.html">
  <TITLE>
    Hydrogen Alpha Image of the Sun Desktop Component
  </TITLE>
  <ABSTRACT>
Continued

This desktop component shows a picture of the Sun
   as it appears this very minute from the top of
   Sacramento Peak in New Mexico. The picture is taken
   in a single color at the wavelength of the Hydrogen
   alpha light (6563 Angstroms) using a monochrome
   camera which produces a greyscale image in
   which the red light of Hydrogen alpha appears white.
  </ABSTRACT>

  <ITEM
    HREF="http://vtt.sunspot.noao.edu/gifs/video/sunnow.jpg">
    <TITLE>Hydrogen Alpha Image of the Sun</TITLE>

    <SCHEDULE TIMEZONE="0700-">
```

```
      <INTERVALTIME MIN="1"/>
      <EARLIESTTIME HOUR="6"/>
      <LATESTTIME HOUR="19"/>
    </SCHEDULE>

    <USAGE VALUE="DesktopComponent">
      <WIDTH VALUE="460"/>
      <HEIGHT VALUE="480"/>
      <CANRESIZE VALUE="yes"/>
      <OPENAS VALUE="Image"/>
    </USAGE>
  </ITEM>
</CHANNEL>
```

قيمة البريد الإلكتروني

من الطبيعي أنه عندما يخبر المستعرض المشترك بتغير في محتوى القناة بواسطة إرسالهم بالبريد الإلكتروني، يرسله خلال الصفحة الرئيسية للقناة مثل نص رسالة البريد الإلكتروني. بينما، يمكنك تحديد انه يمكن إرسال رسالة مختلفة بالبريد الإلكتروني عن طريق تضمين ITEM في القناة التي يكون عنصرها USAGE له قيمة البريد الإلكتروني.

تحدد تعليمات البرمجة ٣٢-٢٣ أن الملف عند http://metalab.unc.edu/xml/what-snew.html سيتم استخدامه في إخبار المشتركين بتغيرات المحتوى إذا لم يكون ITEM الأول موجودة، إذن http://metalab.unc.edu/xml/ من سمة CHANNEL HREF سيتم استخدامها بدلاً منة. وهذا يعطيك الفرصة لإرسال رسالة مختصرة تحدد بالضبط ما تم تغييره، بدلاً من إرسال الصفحة المتغيرة بأكملها. دائماً تكون معلومات What new أسهل للقراء في استيعابها من الصفحة بأكملها.

تعليمات البرمجة ٢١-١٢: القناة التي ترسل بالبريد الإلكتروني ملاحظة مفصلة

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">
  <TITLE>Cafe con Leche</TITLE>
  <ABSTRACT>
    Independent XML news and information for content
    and software developers
  </ABSTRACT>
```

```
<LOGO HREF="cup_ICON.gif" STYLE="ICON"/>
<LOGO HREF="cup_IMAGE.gif" STYLE="IMAGE"/>
<LOGO HREF="cup_IMAGE-WIDE.gif" STYLE="IMAGE-WIDE"/>

<ITEM HREF="whatsnews.html">
 <USAGE VALUE="Email"/>
</ITEM>

<ITEM HREF="books.html">
 <TITLE>Books about XML</TITLE>
 <ABSTRACT>
  A comprehensive list of books about XML
  with capsule reviews and ratings
 </ABSTRACT>
</ITEM>

<ITEM HREF="tradeshows.html">
 <TITLE>Trade shows and conferences about XML</TITLE>
 <ABSTRACT>
  Upcoming conferences and shows with an XML focus
 </ABSTRACT>
</ITEM>

<ITEM HREF="mailinglists.html">
 <TITLE>Mailing Lists dedicated to XML</TITLE>
 <ABSTRACT>
  Mailing lists where you can discuss XML
 </ABSTRACT>
</ITEM>
</CHANNEL>
```

قيمة NONE

لا تظهر البنود التي تكون قيمة USAGE الخاصة بها NONE في أي مكان، لا تظهر في شريط القناة، المفصلة، ولا في أي مكان. بينما، يتم التخزين المؤقت السابق لمثل هذه البنود ولهذا تكون متوافرة بسرعة أكبر لصفحات التطبيقات الصغيرة وHTML التي تشير إليهم فيما بعد.

يعد محتوى القناة الذي تم تخزينه مفيد لتضمين البنود، مثل النقاط والصوت الذي تريد نقلها إلى جهاز القارئ لاستخدامه عن طريق صفحات القناة. يمنك أن تقوم بالتخزين المؤقت السابق لبند فردي أو سلسلة من البنود عن طريق تعريف القناة التي تتضمن مجموعة البنود التي تم تخزينها مؤقتاً سابقاً، كما هو موضح في هذا المثال:

```
<ITEM HREF="welcome.wav"> <USAGE="NONE"/> </ITEM>
<ITEM HREF="spacemusic.au"> <USAGE="NONE"/> </ITEM>
```

يتضمن تلك المثال ملفين للصوت مستخدمين في الموقع عندما يقوم المستعرض بتحميل محتويات القناة للمشاهدة غير المتصلة لا يتم عرض هذين الملفين في شريط القناة، ولكن إذا كان الملف الموجود في شريط القناة يستخدم. واحد من تلك الملفات الصوتية، إذن ستكون متوفرة على الفوز، ومحملة عندما تكون الصفحة مرئية ير متصلة. لا يتحتم على القارئ انتظارهم حتى يتم تحميلها من موقع ويب البعيد، فهناك الوسائط اعتبار مهم أثناء تناول الملفات متعددة الوسائط الكبيرة نسبياً.

قيمة شاشة التوقف

تشير البنود التي تكون قيمة USAGE الخاصة بها ScreenSaver إلى صفحة HTML لتحل محل الشاشة الرئيسية العادية بعد فترة عدم نشاط محدده من قبل المستخدم عامة، يتم كتابه شاشة التوقف كمستند CDF منفصل تماماً عن القناة العادية، وتطالب بتحميل منفصل ورابطة ثانية. على سبيل المثال.

```
<A HREF="ccl_screensaver.cdf">
 Download and install the Cafe con Leche Screen Saver!
</A>
```

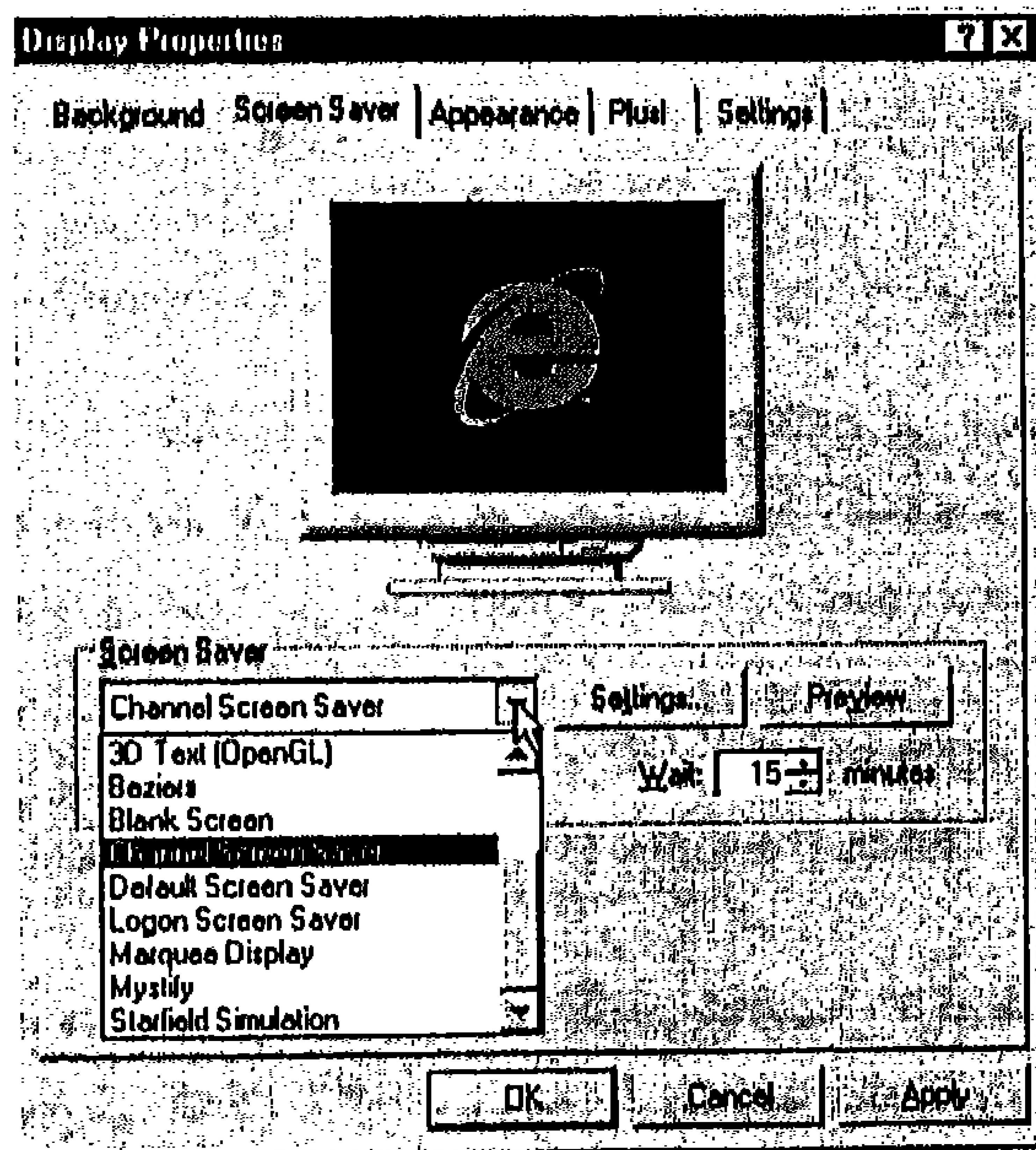
ما لم يكون المشترك قد حدد شاشة توقف القناة كنظام شاشة التوقف في لوح التحكم المعروض كما هو واضح في الشكل ٣٢-٧، سيسأل المستعرض المستخدم عما إذا كانوا يريدون استخدام Channel Screen Saver أو شاشة التوقف القناة، المرة القادمة التي تحفظ فيها الشاشة، فإن المستند الذي سيتم الإشارة أليه في قناة حفظ الشاشة يتم تحمله وعرضه. إذا اشترك المستخدم في اكثر من قناة لحفظ الشاشة، فيدور المستعرض خلال قنوات حفظ الشاشة الخاصة بالمشترك كل ٣١ ثانية. قد يتغير المستخدم تلك الفاصل الزمني وقليل من الاختيارات الأخرى

"إذا كانت شاشة التوقف تقرأ الأصوات على سبيل المثال" مستخدماً إعدادات حفظ الشاشة في لوح التحكم المعروض.

تعد تعليمات البرمجية ٢١-٢٣ قناة شاشة التوقف بسيطة. ويتم الإشارة إلى المستند الفعلي المعروض أثناء حفظ الشاشة عن طريق عناصر ITEM لسمة HREF. ستعمل هذه الصفحة عامة على الاستخدام الثقيل DHTML، JavaScript، والحيل الأخرى لإظهار حركة الشاشة. تعد صفحة شاشة التوقف التابعة فكرة سيئة.

تعليمات البرمجة ٢١-١٣ : قناة شاشة التوقف

```
<?xml version="1.0"?>
<CHANNEL BASE="http://metalab.unc.edu/xml/">
<ITEM HREF="http://metalab.unc.edu/screensaver.html">
   <USAGE VALUE="ScreenSaver"/>
  </ITEM>
</CHANNEL>
```



الشكل ٢١-٧ مفتاح جدول شاشة التوقف للوح التحكم في الخصائص المعروضة في ويندوز NT 4.0

عليك أن تأخذ في اعتبارك شيئان عند تصميم شاشة التوقف:

١- فرضاً أن المستخدم يعمل شيئاً أخر عند حفظ الشاشة بعد كل شئ، ينشط عــدم النشــاط "الكل"، لهذا لا تتوقع كثير من انتباه المستخدم أو التفاعل مع شاشة التوقف الخاصة بك.

٢- على الرغم من أن أغلب العرض غير الحديث يحتاج حقاً إلى حفظ شاشته، لذلك يجــب على شاشات التوقف أن تحفظ الشاشة. ولهذا يجب أن تكون معظم الشاشة مظلمة أغلــب الوقت، ولا يجب على أي بكسل أن يستمر بلون غير- أسود واحد، خاصة الأبيض.

قيمة تحديث البرنامج

تعتبر القيمة الممكنة النهائية لعنصر USAGE هي SoftwareUpdate لا تقتصر القنوات فقــط على تسليم الأخبار وصفحات الويب. ويمكنهم إرسال تحديث البرامج أيضاً. يمكن للقنوات الــذي تم تحديث برنامجها أن تخبر المستخدمين بالتحديث للبرامج وتسليم المنتج عبر الإنترنت. بإعطــاء ثقة كافية للمستخدم، يمكنهم تثبيت البرامج آليا.

لكي تنشئ قناة تقدم للبرنامج، اكتب ملف CDF بجذر عنصر CHANNEL الذي يعد عنصــره USAGE له قيمة تحديث البرنامج. يمكن أن يكون لهذه القناة عنــوان، software، شــعارات، وجدول، تماماً مثل أي قناة أخرى. تعد تعليمات البرمجية ٣٢-٢٤ قناة تحديث البرنامج مزيفة.

تعليمات البرمجة ٢١-١٤ : قناة تحديث البرامج

```
<?xml version="1.0"?>
<CHANNEL HREF="http://www.whizzywriter.com/updates/2001.html">
  <TITLE>WhizzyWriter 2001 Update</TITLE>
  <ABSTRACT>
    WhizzyWriter 2001 offers the same kitchen sink approach
    to word processing that WhizzyWriter 2000 was infamous for,
    but now with tint control! plus many more six-legged
    friends to delight and amuse!  Don't worry though. All the
    old arthropods you"ve learned to love and adore in the
    last 2000 versions are still here!
  </ABSTRACT>

  <USAGE VALUE="softwareUpdate"/>
```

```
<SOFTPKG NAME="WhizzyWriter 2001 with tint control 2.1EA3"
 HREF="http://www.whizzywriter.com/updates/2001.cab"
 VERSION="2001.0,d,3245" STYLE="activeSetup">

 <!- other OSD elements can go here ->

</SOFTPKG>

</CHANNEL>
```

بجانب قيمة عنصر USAGE، يعد المفتاح لقناة تخزين البرنامج هو عنصرها التابع SOFTPKG. توفر سمة HREF لعنصر URL SOFTPKG الذي عن طريق يمكن تحميل البرنامج وتثبيته، يجب أن تشير URL إلى أرشيف مضغوط للبرنامج في (CAB) تنسيق خزانة Microsoft. يجب أن يحمل هذا الأرشيف توقيع رقمي من سلطة مفوضة. مزيد بالإضافة إلى، يجب أن تحتوي أيضاً على ملف OSD الذي يصف تحديث البرنامج. يعد (OSD) التنسيق الوصفي لبرنامج المفتوح تطبيق XML لوصف تحديث البرنامج المبتكر بواسطة Microsoft و.Marimba يتم وصف لغة وبنية ملف OSD على موقع ويب Microsoft. .http://www.microsoft.com/stan-dards/osd/.

المرجع يتم تغطية OSD باختصار في الباب الثاني تحت عنوان، مقدمه لتطبيقات XML.

يجب أيضاً أن يكون لعنصر SOFTPKG سمة NAME التي تحتوي على ما يزيد عن ٢٦٠ حرف يقول بوصف التطبيقات على سبيل المثال، WhizzyWriter 2100 with tint control .2.1EA3

يجب أيضا أن يكون لعنصر SOFTPKG سمة STYLE بواحدة من القمتين ActiveSetup أو MSICD "تحميل مكون الإنترنت Microsoft" الآتي تحدد كيفية تحميل وتثبيت البرنامج.

يوجد أيضا العديد من سمات SOFTPKG الاختيارية. قد يكون لعنصر SOFTPKG سمة PRECACHE إما بالقيمة Yes أوNo ويكون لهذا نفس المعنى مثل سمات PRECACHE الأخرى، وهي تحديد هل سيتم تحميل الحزمة قبل أن يقرر المستخدم إذا كان يريدها ام لا تعد سمة VERSION قائمة مفصولة بفاصلة لها أرقام كبيرة، صغير، مخصص، وإصدار مبنى، مثل ٦،٢،٣،٣١٢٤ أخيراً، يقوم إعداد سمة AUTOINSTALL إلى Yes بأخبار المستعرض أن يحمل

حزمة البرنامج آلياً بمجرد تحميل مستند CDF. ترشيد القيمة No المستعرض لينظر طلب خلاص للمستخدم. وهو افتراض إذا لم يتم تضمين سمة AUTOINSTALL.

يمكن للعناصر التابعة أن تكون داخل عنصر SOFTPKG

- TITLE
- ABSTRACT
- LANGUAGE
- DEPENDENCY
- NATIVECODE
- IMPLEMENTATION

بينما، لا تعتبر تلك العناصر جزء من CDF. ولكن يعتبروا جزء من OSD. "تقنياً تعد SOFTPKG هي أيضا" وبالتالي، سأقوم فقط بتلخيصهما هنا.

- يستخدم عنصر TITLE الخاص SOFTPKG نفس الاختيارات مثل CDF TITLE.
- يصف عنصر ABSTRACT البرنامج وهو بالضرورة نفس عنصر CDF ABSTRACT.
- يعرف عنصر LANGUAGE اللغة المدعمة بواسطة هذا التحديث الذي يستخدم سمة VALUE والذي يعد قيمة ISO 639/RFC 1766 للتعليمات البرمجية للغة المكونة من حرفين مثل EN للغة الإنجليزية. إذا تم تدعم اللغات المتعددة، يتم فصلهم عن طريق الفاصلة المنقوطة.
- يعد عنصر DEPENDENCY فارغ بسمة فردية، وسيأخذ ACTION قيمة واحدة من القيمتين Assert أو Install. تعد Assert افتراضية وتعني أن التحديث سيكون مثبت فقط إذا كان ملف CAB الضروري موجود بالفعل على الكمبيوتر المحلي. أما مع قيمة Install، يتم تحميل الملفات الضرورية من الخادم.
- يحمل عنصر NATIVECODE عناصر CODE التابعة. ويشير كل عنصر CODE تابع إلى ملفات التوزيع من أجل معمار محدد مثل ويندوز ٩٨ على x٨٦ أو ويندوز NT على الألف.
- يصف عنصر IMPLEMENTATION الشكل المطلوب لحزمه البرنامج إذا كانت المطالب الموصوفة في علامة التنفيذ غير موجودة على جهاز القارئ، لن يتقدم التحميل والتثبيت. يعد عنصر IMPLEMENTATION عنصر اختيار بعناصر تابعة مثل CODEBASE, LANGUAGE وOS وPROCESSOR.

يكون لعنصر CODEBASE سمات FILENAME وHREF التي تقول المكان التي توجد فيه الملفات التي سيتم تحديثها.

يعد عنصر LANGUAGE تماما مثل عنصر LANGUAGE السابق.

يكون لعنصر OS سمة VALUE التي تكون قيمتها Mac, Win95 أو Winnt. ولذلك فهي تعرف نظام التشغيل المطلوب للبرنامج. يجوز أن يكون لهذا العنصر تابع فارغ يسمى OSVERSION بسمه VALUE الخاصة به التي تعرف الإصدار المطلوب.

يعد عنصر PROCESSOR عنصر فارغ تستطيع سمة VALUE الخاصة به أن يكون لها قيمة .Alpha, MIPS, PPC, or x86. يصف هذا معمار CPU الذي يدعمه البرنامج.

للحصول على مزيد من التفاصيل عن OSD، يمكنك أن ترى الإشارة عند http://www.micro-soft.com/workshop/delivery/osd/reference/reference.asp, أو المواصفات عند .http://www.microsoft.com/standards/osd/default.asp

خلاصة

لقد تعلمت في هذا الباب:

- يعد التنسيق المعروف للقناة (CDF) تطبيقات XML المستخدمة لوصف البيانات المقدمة من مواقع ويب إلى مستعرض ويب.
- تعد ملفات CDF ملفات CDF تطبيقات XML المستخدمة لوصف البيانات المقدمة من مواقع ويب إلى مستعرض ويب.
- تعد ملفات CDF ملفات XML، على الرغم من أنهم عندهم على وجه مخصص أسم ملحق الملف xml. بدلاً من cdf. ويعد جذر عنصر ملف CDF هو CHANNEL يجب أن يحتوي كل عنصر CHANNEL على سمة HREF التي تعرف الصفحة المقدمة.
- قد يحتوي عنصر CHANNEL على عناصر ITEM التابعة الإضافة التي تحتوي سمات URLs الخاصة بها على URLs ويكون لها صفحات إضافية تقدم.
- يحتوي كل عنصر ITEM وCHANNEL على توابع ABSTRACT ,TITLE و LOGO التي تصف محتوى الصفحة وإشارات العنصر.
- يحدد عنصر SCHEDULE متى وكم مرة يجب أن يفحص فيها المستعرض الخادم لتحديثه.
- يعرف عنصر LOG البنود التي يرجع تقرير مظهرها إلى خادم الويب، على الرغم من أن المشترك قد يعطل هذا التقرير.

- يعرف عنصر LOGTARGET كيف يتم إرجاع معلومات تسجيل الدخول من القناة إلـــى الخادم.
- يوفر سمة BASE نقطة بداية التي يتم من خلالها حساب URLs النســـبي فـــي سـمات HREF للعنصر التابع.
- تحدد سمة LASTMOD الوقت الأخير الذي قد تم تغير الصفحة فيه، حتى يســـتطيع أن يقول المستعرض هل يحتاج أن يحمل أم لا.
- تسمح سمة USAGE لك باستخدام صفحات ويب كقنوات، محتوى تخزينه موقتاً، مكونات الشاشة الرئيسية النشطة، شاشات التوقف، وتحديث البرنامج.

مستكشف الباب القادم تطبيقات XML مختلفة تماماً لتوجيه الرسومات لغة الترميز المحوريـــة Vector Markup Language (VML).



الفصل

لغة الترميز المحورية

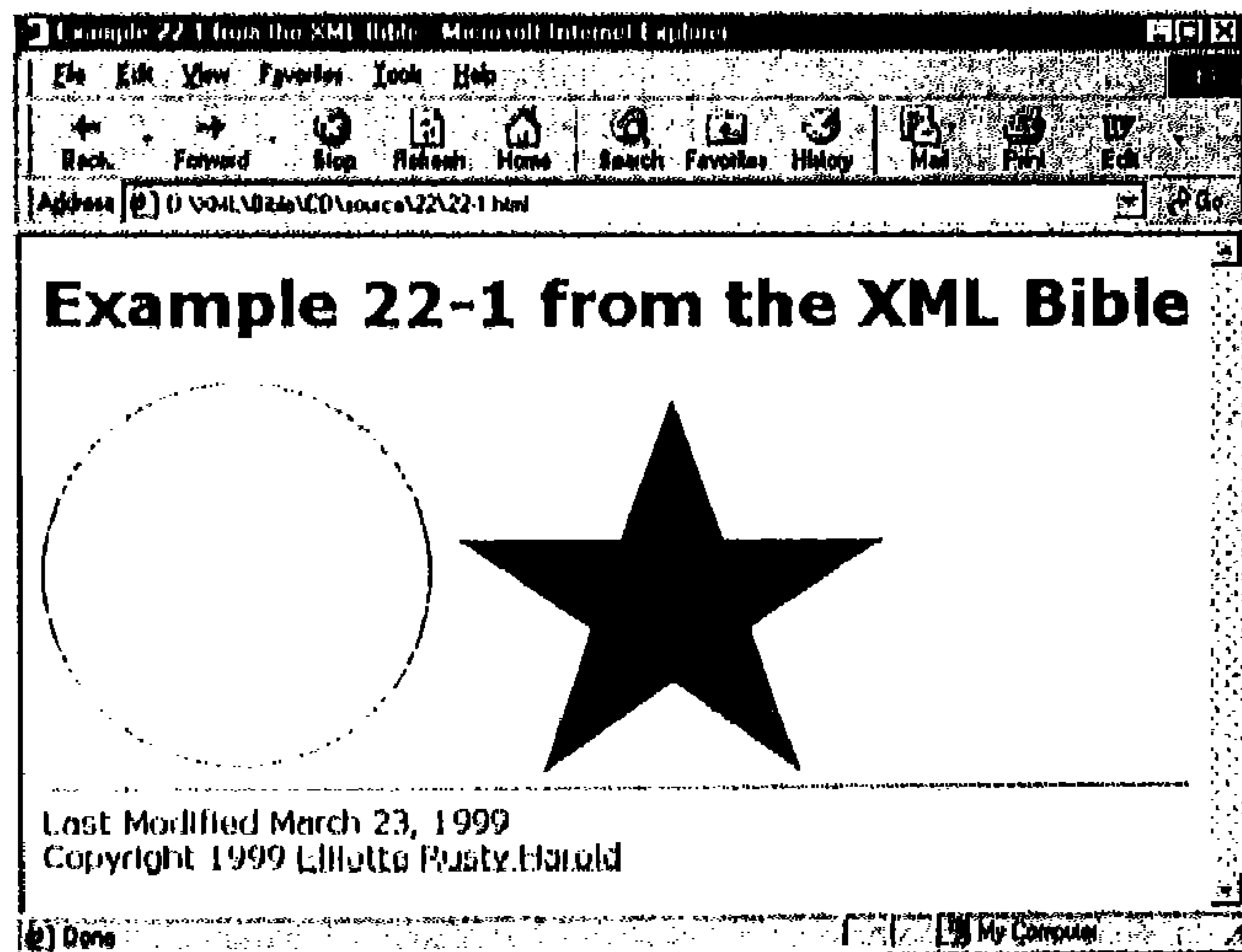
في هذا الفصل

موضوع هذا الفصل هو لغة الترميز المحورية، التي هي إحدى تطبيقات XML التي تجمع معلومات اتجاهيه مع ترميز CSS لوصف صور بيانية اتجاهيه يمكن تضمينها في صفحات ويب محل الصور النقطية GIF وJPEG التي يحملها IMG، عنصر HTML فالصور البيانية الاتجاهية تشغيل حيزاً أقل وبالتالي تقوم بالعرض على اتصالات الشبكة البطيئة أسرع من الصور النقطة التقليدية GIF وJPEG ويدعم VML المكونات المختلفة لبرنامج مايكروسوفت أوفيس ٢٠٠٠ "وورد، باور بوينت وإكسل" إلى جانب Internet Explorer 5.0 فعند حفظ مستند وورد ٢٠٠٠ أو باور بوينت ٢٠٠٠ أو إكسل ٢٠٠٠ تحت اسم HTML، فإن الصور البيانية الموجودة في تلك البرامج تتحول إلى VML.

ماذا تعني VML

تمثل عناصر VML أشكالاً مثل: المستطيلات، والأشكال البيضاوية، والدوائر، والمثلثات، والسحب وشبه المنحرف إلى غير ذلك من الأشكال. ويوصف كل شكل بأنه مسار يتكون من سلسة من الخطوط المتصلة والمنحنيات. وتستخدم VML العناصر والسمات لوصف المخطط التفصيلي والتعبئة والاتجاه وغيرها من الخصائص الخاصة بكل شكل. ويمكن تطبيق خصائص CSS القياسية علي عناصر VML لتعيين اتجاهاتها.

تعليمات البرمجة ٢٢-١ هي مستند HTML، وبداخل ملف HTML توجد تعليمات البرمجة VML لرسم نجمة زرقاء ذات خمسة أطراف ودائرة حمراء والشكل ٢٢-١ يوضح المستند المعروض في Internet Explorer 5.0.



الشكل ٢٢-١ مستند HTML متضمن عناصر VML

تعليمات البرمجة ٢٢-١ : مستند HTML يحتوي على تعلمته برمجة VML لرسم نجمة زرقاء ذات خمسة أطرف ودائرة حمراء

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

  <head>
    <title>Example 22-1 from the XML Bible</title>
    <object id="VMLRender"
      classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
<style>
      vml\:* { behavior: url(#VMLRender) }
    </style>
  </head>

  <body>
    <h1>Example 22-1 from the XML Bible</h1>

    <div>
      <vml:oval
        style="width:200px; height: 200px"
        stroke="true"
        strokecolor="red"
        strokeweight="2">
      </vml:oval>

      <vml:polyline
        style="width: 250px; height: 250px"
        stroke="false"
        fill="true"
        fillcolor="blue"
        points="8pt, 65pt, 72pt, 65pt, 92pt, 11pt, 112pt, 65pt,
                174pt, 65pt, 122pt, 100pt, 142pt, 155pt, 92pt,
                121pt, 42pt, 155pt, 60pt, 100pt">
      </vml:polyline>
    </div>
    <hr></hr>
    Last Modified March 23, 1999<br />
```

```
    Copyright 1999
    <a href="http://www.macfaq.com/personal.html">
      Elliotte Rusty Harold
    </a>
  </body>

</html>
```

لا يمكن اعتبار تعليمات البرمجة ٢٢-١ مستند HTML عادياً على الرغم من احتوائه علـــى بعض عناصر HTML القياسية فأولاً، يعلن العنصر الجذري html بـــادئ مســافة الاســم vml كاختزال urn:schemas-microsoft-com:vml ويحتوي العنصر head على نتيجــة object مع VMLRender" id VMLRender" هو برنامج مثبت "IE5 وهناك أيضا قاعدة نمط CSS التي تحدد إن كل العناصر في مسافة الاسم vml "أي كل العناصر التي تبدأ vml" يجب أن يكون لــها خاصية url(#VMLRender) behavior وتعد هذه URL نسبية تشـــير إلـــى عنصـــر object المذكور سابقاً، فيقوم ذلك بأخبار عارض ويب بتمرير كل العناصر التي لها بادئة vml "يســتخدم الخط المائل العكسي في :\\vml ليؤكد على أن: تعامل على أنها جزء من اسم العنصر وليس على أنها فاصل مختار" إلى الكائن الذي له الرقم المعروف VMLRender ليصبح جاهزاً للعرض.

أما عن عنصر body، فهو يحتوي على العديد من عناصر HTML المعتادة مثـــل div وh1 و hr, وa.، كما تحتوي أيضاً على عناصر vml:oval وvml:polyline يعين عنصـــر vml:oval ليكون له حد احمر اللون عرضه ٢بسكل. إضافة إلى ذلك تعيـــن ســـمة style خـــاصتي CSS width وheight لهذا البيضوي، بمائتي بكسل لكل منهما. وتعين تلك الخواص أيضا عـــرض وارتفاع الصندوق المستطيل الضمني الذي يحمل البيضاوي وعنصر vml:polyline معد لأن يتم ملأه باللون الأزرق بينما تعد خاصيتي width وCSS height لهذا البيضاوي لأن تكـــون لكـــل منهما ٢٥٠ بكسل وبما أنه النجمة ذات الخمسة أطرف عشرة قمم، فإن سمة properties توفـــر عشرة أزواج من روابط س-ص، حتى يكون هناك زوج واحد لكل قمة.

الرسم باستخدام لوحة المفاتيح

يعد رسم الصور باستخدام لوحة المفاتيح تماماً كدق مسمار في قطعة من الخشب باستخدام قطعــة إسفنج. فكتابة صور VML عن طريق طبع تعليمات برمجة أولية من نوع XML في محرر النص ليس بالشيء السهل. لذا اقترح عند بدء أي محاولات لبرمجة صور اتجاهيه، إن تستعين بـــورق رسم بياني وترسم الصور بالطريقة التي تود رؤيتها بها علي الشاشة، باستخدام القلم الرصـــاص. ومن ثم يمكنك بعد ذلك استخدام الصور المرسومة على ورق الرســم البيـــاني لتحديـــد روابـــط عناصر VML المتعددة مثل shape وoval وpolyline.

عنصر الشكل

يعتبر عنصر board عنصر VML الرئيسي، وهو يقوم برسم منحني تحكمي مغلق في اتجـاهين. ومعظم الأشكال يكون لها مساراً يضع حدود الشكل، وهذه الحدود قد يكون أولا يكون لها لـــون وعرض محدد، أي أنها قد تكون مرئية أو غير مرئية فعلى سبيل المثال نجد إن الدائرة في الشكل ٢٢-١ لها حدود سوداء لكن بلا لون من الداخل بينما نجد أن النجمة ليست لها حـدود خارجيـة لكنها معبأة من الداخل باللون الأزرق.

يمكن تحديد معظم خواص عنصر shape بسمات متعددة، يوضحها الجـدول ٢٢-١ فعلـى سبيل المثال هاهو عنصر shape يرسم مثلث متساو الساقين. فتعطي سمة id الشكل اسما فريداً، بينما تحدد سمة coordsize حجم نظام الربط الموضعي، وتستخدم سمة style خـاصيتى CSS width وheight لتحديد عرض وارتفاع الصندوق الذي يحتوي على المثلث أما عن سـمة path فهي تقدم البيانات للعنصر الناتج formulas والذي يحسب بقية حدود هذا المثلث وأخيراً تجعـل سمة fillcolor لون المثلث أزرقاً.

```
<vml:shape id="_x0000_t5" coordsize="21600,21600"
 style="width:200px; height: 200px" adj="10800"
 path="m@0,0l0,21600,21600,21600xe" fillcolor="blue">
  <vml:formulas>
   <vml:f eqn="val #0"/>
   <vml:f eqn="prod #0 1 2"/>
   <vml:f eqn="sum @1 10800 0"/>
  </vml:formulas>ss
</vml:shape>
```

وإذا لم تستطع التوصل إلى أن هذا الشكل هو مثلث متساوي الساقين، فلا تقلق، ففي الواقـع التواصل إلى إن هذا الشكل هو مجرد مثلث يدهشني. فمعظم عنـاصر VML "بمـا فيـها هـذا العنصر" يتم رسمها بواسطة GUI وحفظها في شكل VML وتبعاً لذلك لن تكون بحاجة لمعرفـة بناء الجملة المفصل لكل عنصر أو سمة VML لكن لو كنت تعرف القليل فقـط، فإنـه يمكنـك استخدام بعض الحيل المدهشة مع ملف VML والتي يستحيل استخدامها مع المحرر الرسـومي، حيث يمكنك مثلاً البحث عن كل العناصر الزرقاء اللون وتحويلها إلى اللون الحمر.

الجدول ٢٢-١

سمات عنصر الشكل (Shape)

السمة	القيمة الافتراضية	الوصف
Id	لا يوجد	اسم XML فريد للعنصر تماماً مثل أي سمة XML ذات نوع ID.
Adj	لا يوجد	معلمات إدخال للعناصر الناتجة formulas التي تحدد مسار الشكل.
Alt	لا يوجد	نص بديل يظهر في حالة عدم إمكانية رسم الشكل لأي سبب من الأسباب، أمثل سمة ALT لعنصر HTML الذي يحمل اسم IMG.
Chromakey	لا يوجد	خلفية للشكل شفافة اللون، تظهر كل شئ خلف الشكل، مثل: red و#66FF33.
Class	لا يوجد	فئة CSS الخاصية بالشكل.
Coordorigin	0 0	ركن ربط موضعي "محلي" للركن العلوي الأيسر لصندوق الشكل.
Coordsize	1000 1000	عرض وارتفاع صندوق الشكل في مسافة الربط الموضعي.
Fill	True	إذا كان الشكل معبأ.
Fillcolor	White	اللون المعبأ به الشكل مثل red و#66FF33
Href	لا يوجد	URL الانتقال إلية عند النقر على الشكل.
Opacity	1.0	شفافية الشكل كرقم للفاصلة العائمة بين ٠.٥ غير مرئي، و1.0 كامد تماماً.
Path	لا يوجد	أوامر تحديد مسار الشكل.
Print	True	ما إذا كان سيتم طبع الشكل عند طبع الصفحة.

الجدول ٢٢-١		
سمات عنصر الشكل (Shape)		
Stroke	True	ما إذا كان يجب رسم مسار "حدود" الشكل.
Strokecolor	Black	اللون المستخدم لرسم مسار الشكل.
Strokeweight	0.75pt	عرض الخط المستخدم لرسم مسار الشكل
Style	لا يوجد	خواص CSS التي تطبق على هذا الشكل.
Target	لا يوجد	اسم الإطار المحمل عندما تقوم صفحة إعداد الإطار بالتحميل.
Title	لا يوجد	اسم الشكل
Type	لا يوجد	مرجع لك id الخاص بعنصر shapetype.
V	لا يوجد	أمر يقوم بتحديد مسار الشكل.
Wrapcoords	لا يوجد	يحدد كيف يلتف النص باحكام حول shape

في بعض الأحيان يكون أنسب بعض خواص الأشكال أن يتم تعينها مع العناصر الناتجة بدلا من السمات إضافة إلى ذلك تسمح العناصر الناتجة بتحكم أكثر تفصيلا لبعض الجوانب الخاصة بالشكل. فعلى سبيل المثال، المثلث متساوي الساقين المذكور سلفاً، تطلب ثلاث صيغ لوصف المسار، وكل صيغة غلفت في العنصر الناتج vml:f ومن ناحية أخرى نجد أنه باستخدام السمات لا يمكن تضمين غير صيغة واحدة والجدول ٢٢-٢ يضع قائمة بالعناصر الناتجة الممكنة لشكل ما، وإذا حدث إن تعارض العنصر الناتج مع سمة من السمات، فإنه يتم استخدام القيمة التي يحددها العنصر الناتج:.

الجدول ٢٢-٢	
العناصر الناتجة للشكل	
العنصر	الوصف
path	جانبية الشكل.
Formulas	الصيغ التي تحدد المخطط التفصيلي للشكل.

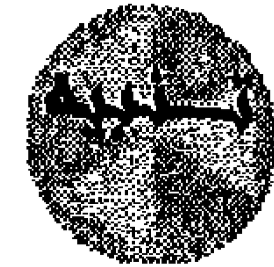
الجدول ٢٢-٢

العناصر الناتجة للشكل

العنصر	الوصف
Handles	عناصر تحكم مرئية تستخدم لتغيير الشكل.
Fill	كيفية تعبئة المسار.
Stroke	كيفية رسم المسار، وذلك في حالة الحاجة إلى شئ أكثر تفصيلا من مجرد خط مستقيم ولون خالص.
shadow	تأثير الظل الخاص بالشكل.
Textbox	النص الذي يجب ظهوره داخل الشكل.
Textpath	المسار الاتجاهي الذي يستخدمه النص.
Imagedata	صورة تؤدى على قمة الشكل.
Line	مسار على هيئة خط مستقيم.
Polyline	مسار يتم تحديده عن طريق توصيل النقاط بين نقاط محددة
Curve	مسار يحدده منحي Bezier مكعب.
Rect	مسار يحدده مستطيل له ارتفاع وعرض محددان.
Roundrect	مسار يحدده مستطيل له أركان مستديرة لها حجم محدد
Oval	مسار يحدده شكل بيضوي ينطوي علية مستطيل له ارتفاع وعرض محددان.
Arc	مسار يحدده قوس لزاوية بين نقطتين.
Image	صورة نقدية مخلية من مصدر خارجي.

يحتوي كل عنصر من هذه العناصر على مجموعة متنوعة من السمات والعناصر الناتجة لتحديد شكله الخارجي. فعلى سبيل المثال، عنصر line، أحد أبسط العناصر، يحتوي على سمات fromو to التي تحدد نقاط البداية والنهاية للخط، وتكون قيمة كل من هذه السمات رابط 2-D في مسافة الربط الموضعية مثل 5 0 و10 32 يمكنك معرفة تفاصيل اكثر من خلال خضوع لغة

الترميز المحورية W3C، أو من على موقع مايكروسوفت أوفيس على .http://www.microsoft.com/ standards/vml/.



كن حذر عند التعامل مع مواصفة VML، فكما تحتوي على الكثير من الأخطاء الواضحة فأنها تحتوى أيضا على لأبأس به من الأخطاء غير الواضحة.



ليس من الضروري أن تكون العناصر line وpolyline وcurve وrect وroundrect وoval وarc وimag.

عنصر نوع الشكل

يحدد عنصر shapetype الشكل الذي يمكن إعادة استخدامه مرات عدة، وذلك بالرجوع أليه عند نقطة معينة في المستند باستخدام عنصر shape ويعد عنصر shapetype مماثلاً لعنصر Shope في كل شئ، غير أنه لا يمكن استخدامه لذكر عنصر shape فهو يشير إلى عنصر أخر، كما أنه دائماً يكون مستتراً. أما عن عنصر shape فهو يشير إلى عنصر shapetype مستخدماً سمة type تكون قيمتها URL نسبي يشير إلى id لعنصر shapetype.

فعلي سبيل المثال تشتمل تعليمات البرمجة ٢٢-٢ عنصر shapetype الذي يحدد مثلث أزرق مستقيم، كما تشتمل أيضاً على ثلاثة عناصر أشكال والتي تذكر فقط عنصر shapetype هذا. لذا نجد ثلاثة مثلثات مستقيمة في الشكل ٢٢-٢ على الرغم من أنه تم تعريفها مرة واحدة فقط. ولكل من هذه المثلثات حجم مختلف من تعيينه في عناصر shope المستقلة على الرغم من أنها جميعاً محسوبة من نفس الصيغ.

تعليمات البرمجة ٢٢-٢ : عناصر shope متعددة تنسخ عنصر shapetype

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

  <head>
   <title>Example 22-2 from the XML Bible</title>
   <object id="VMLRender"
    classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
   </object>
   <style>
```

```
    vml\:* { behavior: url(#VMLRender) }ts
   </style>
  </head>

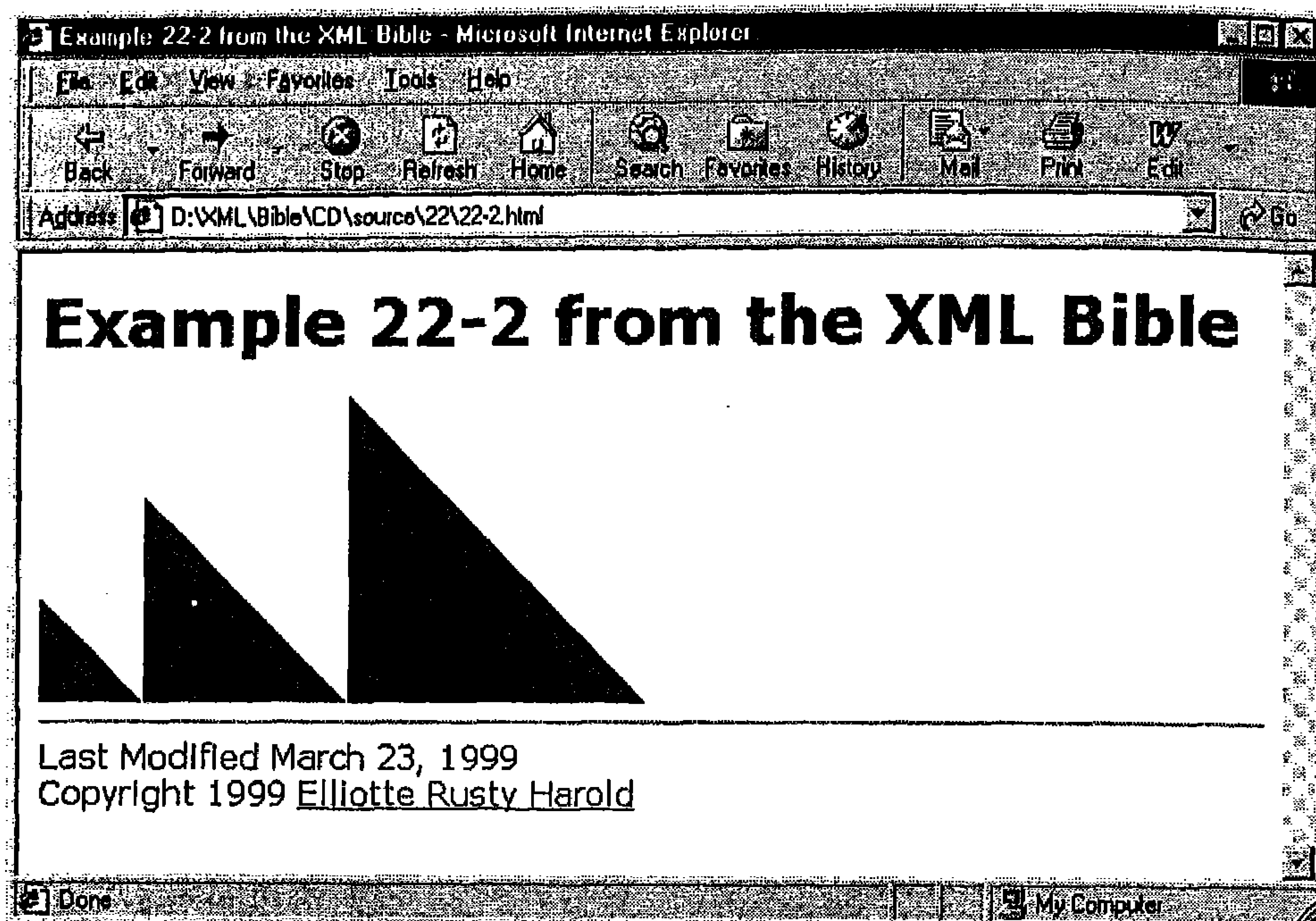
  <body>
   <h1>Example 22-2 from the XML Bible</h1>

   <vml:shapetype id="fred"
coordsize="21600,21600"
      fillcolor="blue"
      path="m@0,0l0,21600,21600,21600xe">
     <vml:formulas>
       <vml:f eqn="val #0"/>
       <vml:f eqn="prod #0 1 2"/>
       <vml:f eqn="sum @1 10800 0"/>
     </vml:formulas>
   </vml:shapetype>

   <vml:shape type="#fred" style="width:50px; height:50px" />
   <vml:shape type="#fred" style="width:100px; height:100px"/>
   <vml:shape type="#fred" style="width:150px; height:150px"/>

   <hr></hr>
   Last Modified March 23, 1999<br />
   Copyright 1999
   <a href="http://www.macfaq.com/personal.html">
     Elliotte Rusty Harold
   </a>
  </body>

</html>
```



الشكل ٢٢-٢ المثلثات المنسوجة من عنصـــر shapetype واحـــد.

عندما يذكر عنصر shape عنصر shapetype، فإنه يمكن لعنصر shape أن يقوم بنســخ بعض السمات التي تطبق على عنصر shapetype وفي هذه الحالة، ستجاوز القيـــم الخاصـــة بعنصر shapetype.

عنصر group

يضم عنصر group الأشكال وغيرها من العناصر الرفيعة المستوى ولعنصـــر group مســافة الربط الموضعية الخاصة به، والتي يكون بها الأشكال الناتجة للعنصر. ثم يمكن بعد ذلك تحريـك هذه المجموعة من الأشكال ووضعها كوحدة. ويمكن أن يشمل عنصـــر group فقطـك علـى السمات الأساسية التي يشمل عليها أي شـــكل(مثــل id وclass وstyle وtitle وhref وtarget وalt وcoordorigin وcoordsize). مثال: يمكنك عمل شكل يتكون من نجمة داخل دائرة بدمج عنصر oval مع polyline في عنصر group على النحو التالي:.

```
<vml:group style="width:6cm; height: 6cm"
 coordorigin="0 0" coordsize="250 250">
 <vml:oval style =
"position:absolute; top: 15; left: 15; width:200; height: 200"
```

```
   stroke="true" strokecolor="black" strokeweight="2"
   fill="true" fillcolor="red">
  </vml:oval>

  <vml:polyline style =
   "position:absolute; top:25; left: 25; width:200; height:200"
   stroke="true" strokecolor="black" strokeweight="5"
   fill="true" fillcolor="blue"
   points="8, 65, 72, ,
        100, 142, 155, 92, 121, 42, 155, 60, 100س>
  </vml:polyline>
 </vml:group>
```

تحدد سمات coordsize وcoordorigin نظام الربط الموضعي للعناصر التي تحتوي عليها المجموعة. فسمة coordorigin تحديد عدد الوحدات الموجودة في موازاة عرض الكتلة الحاوية بينما تحدد سمة coordorigin رابط الركن الأيسر العلوي للكتلة الحاوية.

ويعد هذا النظام نظاما تجريديا، فهو لا يستند إلى أية وحدات مادية مثل البوصة أو البكسل ويعتمد التحويل بين الوحدات المحلية والعالمية على ارتفاع وعرض المجموعة. مثلاً، في المثال السابق. نجد أن كلاً من العرض والارتفاع الحقيقي للمجموعة يبلغ ستة سنتيمترات، وقيمته coordsize is 250 250 ومن ثم تكون قيمة كل وحدة محلية 0.024 (٦سم\٢٥٠). وعندما يتغير الارتفاع والعرض لعنصر contents ستتغير أيضاً بنفس النسب.

يتم اعتبار خواص CSS الموجودة بداخل عنصر group والتي تستخدم لوضع VML مثل leftوwidth، يتم اعتبارها غير بعدية في مسافة الربط المحلية. وبطريقة أخرى يمكن القول بأنها ليست مثل خواص CSS العادية، فهي لا تستخدم الوحدات و وما هي إلا أرقاماً صافية، وليست أطوالاً حقيقية. مثال على ذلك هذا العنصر group:.

```
 <vml:group style="width: 400px; height: 400px"
      coordsize="100.100"
      coordorigin="-50,-50">
```

وعن الكتلة الحاوية، يمكن القول بان عرضها ٤٠٠ بكسل وارتفاعها أيضا ٤٠٠ بكسل وتوضح خاصية coordsize أنة يوجد مائة وحدة أفقية ورأسية بداخل هذه المجموعة، ويبلغ طول كل وحدة محلية ٤ بكسل. ويتراوح نظام الربط داخل الكتلة الحاوية بين 50.0 و-50.0 بمحاذاة المحور 2 وبين أيضا 50.0 و-50.0 بمحاذاة المحور y، و0.0 و0.0 يميناً في مركز

المستطيل والأشكال التي تخذ موضعاً خارج هذه المنطقة، لن تقطع بل ستوضع إلى أعلى أو إلى اسفل العناصر الأخرى في الصفحة. يتم تحديد مكان وحجم كل نتائج العنصر group تبعاً لنظام الربط الموضعي.

وضع أشكال VML مع خواص ورقة نمط متتالية

تناسب عناصر VML المستوى الثاني CSS لنموذج الأداء المرئي تماماً مثل عناصر HTML، وهذا يعني أن كل عنصر VML يكون مضمناً في صندوق يوضح في مكان معين بالصفحة والخواص CSS القياسية التالية تضع الصندوق في مواضع مطلقة أو متغيرة في الصفحة:

- display
- position
- float
- clear
- height
- width
- top
- bottom
- left
- right
- order
- margin
- visibility
- z-index



يناقش كل من الفصل ١٢ "أوراق نمط المستوى ١" خصائص CSS بالإضافة إلى تدعيم نموذج الأداء المرئي القياسي CSS، فإن VML يضيف أربعة خصائص أخرى حتى يمكن للأشكال الاستدارة والانعكاس والوضع في مكان معين، وهذه الخصائص هي:

- rotation
- flip
- center-x
- center-y

ملحوظة

أنا شخصيا اعتقد أن إضافة خصائص CSS غير قياسية لسمة style يعد فكـــرة سيئة للغاية فأنا افضل إن تضاف تلك الخصائص التي قد تسبب مشـــاكل وذلـك لأنهم إلا يفعلان شيئا إضافيا عما تفعله خاصتي left وright.

وتستخدم عناصر VML سمة style لا عداد تلك الخصائص، تماما مثلمـــا تفعـل عنـاصر HTML، ومن ثم يكون لها نفس بنية سمة HTML style. فعلى سبيل المثال هذا الشكل البيضاوي من نوع VML يستخدم سمته style لتعين موضعه وحدوده وخصائص هوامشه:

```
<vml:oval style="top: 15; left: 15; width:200; height: 100;
  margin:10; border-style:solid; border-right-width: 2;
  border-left-width: 2; border-top-width: 1.5;
  border-bottom-width: 1.5"
  stroke="false" fill="true" fillcolor="green">
</vml:oval>
```

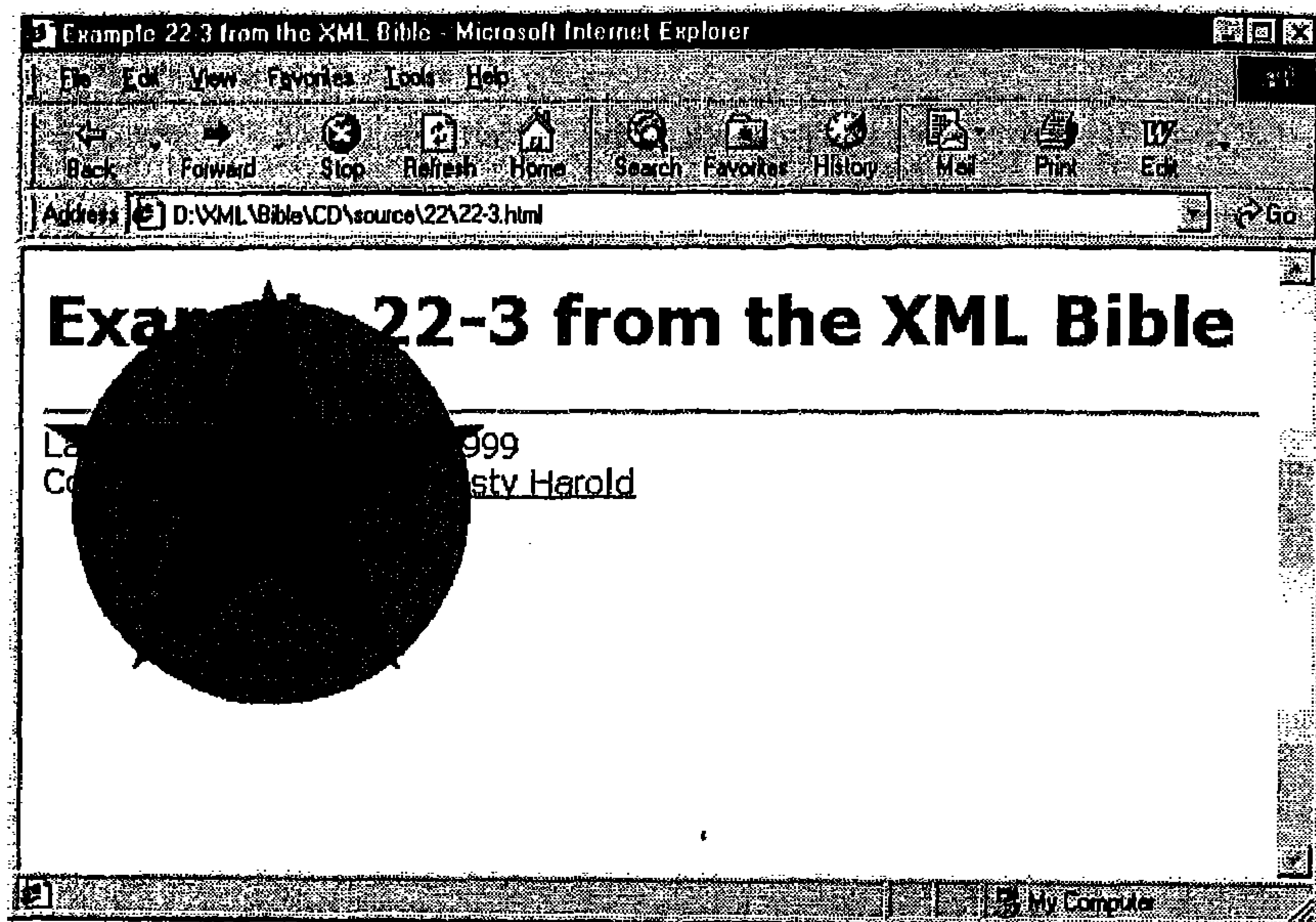
ويتم وضع الأشكال على الصفحة باستخدام خصائص CSS position, left, right height.

وإذا كان لخاصية position property قيمة absolute سيتم الصندوق المستطيل الشكل غـــير المرئي الذي يحتوي على الشكل، يتم وضعة في روابط تتصل بالنافذة التي تعـــرض الصفحـــة، بغض النظر عن أي شئ أخر يظهر في الصفحة، وهذا يعني أن الأشـــكال المختلفـــة وعنـــاصر HTML قد تتداخل. ثم يستخدم VML خاصية CSS z-index لصف العناصر من أسفل إلى أعلى، فتحجب العناصر العلوية مثيلاتها السفلية، وكل ذلك يمكنك من رض العناصر فوق بعضها لبنـــاء صور معقدة لاستخدامها في صفحات ويب. وإذا لم يكن للعنـــاصر خصـــائص z-index، فـــإن العناصر التي تأتي أخيرة توضع فوق التي سبقها في المستند.

تستخدم تعليمات البرمجة ٢٢-٣ تنظيم مطلق لوضع نجمة زرقاء فوق دائرة حمراء، والتـــي هي بالفعل تكون فوق رأس الصفحة h1 وكتلة التوقع، ويوضح الشكل ٢٢-٣ نتيجة ذلك.

والقيمة الافتراضية لخاصية position هي static، وهذا يعني أن كلاً من عنـــاصر HTML وأشكال VML تفسر الواحدة تلو الأخرى، على أن تأخذ كلاً منها المسافة التي تحتاجها، لكن دون صفهم فوق بعضهم البعض.

وعن خاصيته property فهي أيضاً يمكن إن يبدأ العمـــل بــها relative، والتـــي بوضـــع الصندوق الذي يفترض أن تكون به ثم نقلها من هذا الموضع باستخدام الكميـــات المحـــددة فـــي خصائص top و bottom و left و right.



الشكل ٢٢-٣ نجمة زرقاء فوق دائرة حمراء فوق نـــص الصفحـــة.

تعليمات البرمجة ٢٢-٣: تعليمات VML برمجة ترسم نجمة زرقاء ذات خمسة أحرف فوق دائرة حمراء

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

 <head>
  <title>Example 22-3 from the XML Bible</title>
  <object id="VMLRender"
   classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
  </object>
  <style>
   vml\:* { behavior: url(#VMLRender) }
  </style>
```

```
</head>
<body>
  <h1>Example 22-3 from the XML Bible</h1>

  <div>
   <vml:polyline
    style="position:absolute; top:0px; left:0px;
         width: 250px; height: 250px; z-index: 1"
    stroke="false"
    fill="true"
    fillcolor="blue"
    points="8pt, 65pt, 72pt, 65pt, 92pt, 11pt, 112pt, 65pt,
          174pt, 65pt, 122pt, 100pt, 142pt, 155pt, 92pt,
          121pt, 42pt, 155pt, 60pt, 100pt">
   </vml:polyline>

   <vml:oval style="position:absolute; top:25px; left:25px;
              width:200px; height: 200px; z-index: 0س
    stroke="false"
    fill="true"
    fillcolor="red">
   </vml:oval>

  </div>
  <hr></hr>
  Last Modified March 23, 1999<br />
  Copyright 1999
  <a href="http://www.macfaq.com/personal.html">
   Elliotte Rusty Harold
  </a>
 </body>

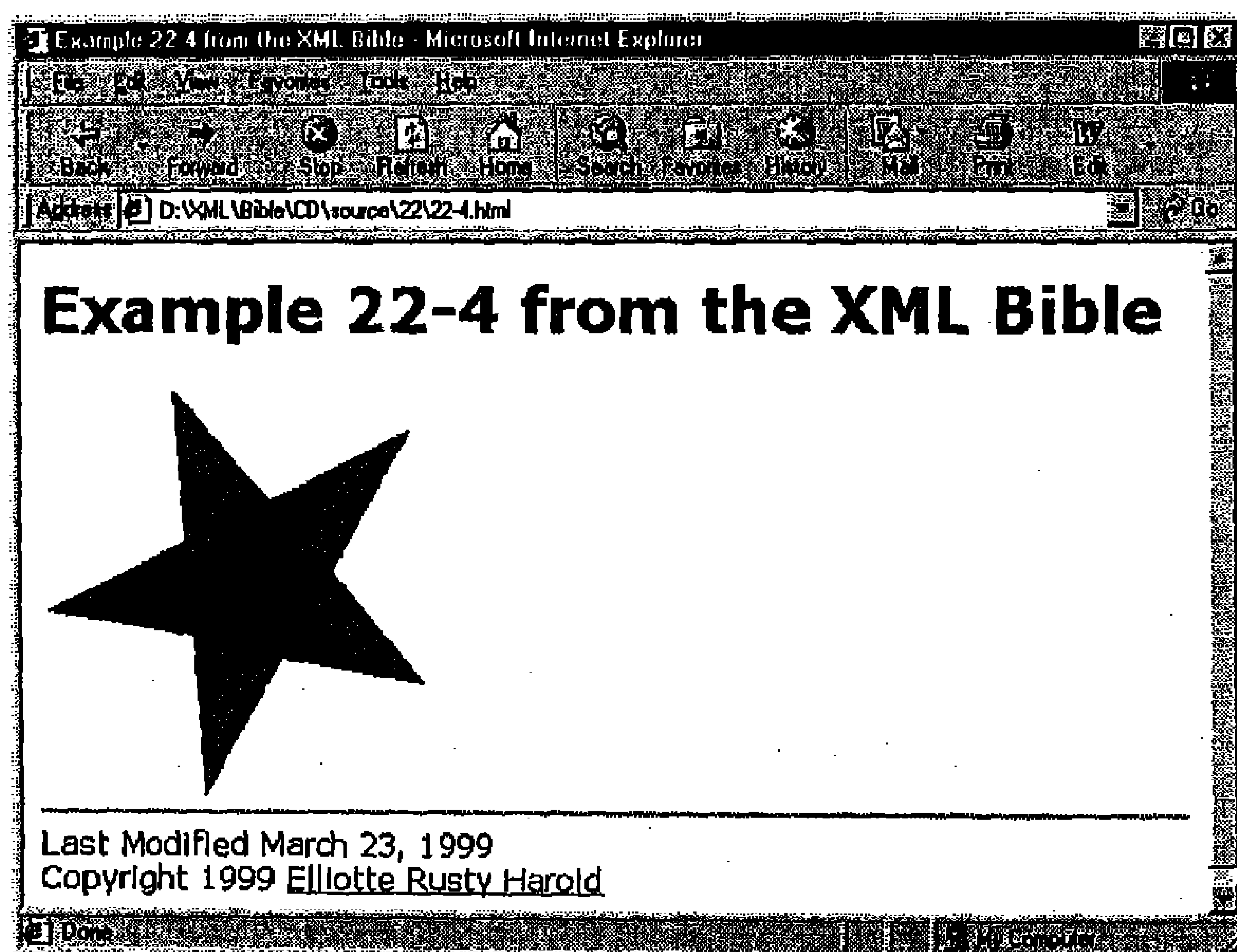
</html>
```

خاصية الدوران

لا يوجد خاصية rotation في قياس CSS لكن يمكن استخدامها كخاصية CSS لأشـــكال VML. وتمثل قيمة خاصية property رقما ما يعبر عن درجة دوران الشكل باتجـــاه حركـــة عقـــارب الساعة حول مركز الشكل. فإذا كان الرقم سالباً فهذا يعني أن الكائن يدور عكس اتجـــاه حركـــة عقارب الساعة. ويتم تحديد القيم بالدرجات فـــي تنسـيقات مثـل 45deg و90deg و-30deg وغيرها. والنجمة المذكورة في تعليمات البرمجة ٢٢-٢ تدور تعليمات البرمجة ٢٢-٤ بمقـــدار ١٢٠ درجة، ويوضح الشكل ٢٢-٤ الشكل النهائي لذلك.

خاصية الانعكاس

تماماً مثل خاصية property، نقول أيضاً أن خاصية flip لا يوجد قياس CSS لأشـــكال VML، فهي تقوم بعكس الشكل حول المحور السيني أو حول المحور الصادي وأو حول المحورين معـــاً، وذلك يعتبر خاصية CSS على سمة style لإحدى عناصر شكل VML ولعكس روابط من حـــول المحور، طبق flip على، وكذلك لعكس روابط حول المحور العادي، طبق flip على س. تعليمات البرمجة ٢٢-٥ تعكس الشكل حول المحور السيني،، ويوضح الشكل ٢٢-٥ الشكل النهائي لذلك.



الشكل ٢٢-٤ نجمة تدور بمقدار ١٢٠ درجـــة

تعليمات البرمجة ٢٢-٤ : نجمة تدور بمقدار ١٢٠ درجة

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

 <head>
  <title>Example 22-4 from the XML Bible</title>
  <object id="VMLRender"
   classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
  </object>
  <style>
   vml\:* { behavior: url(#VMLRender) }
  </style>
 </head>

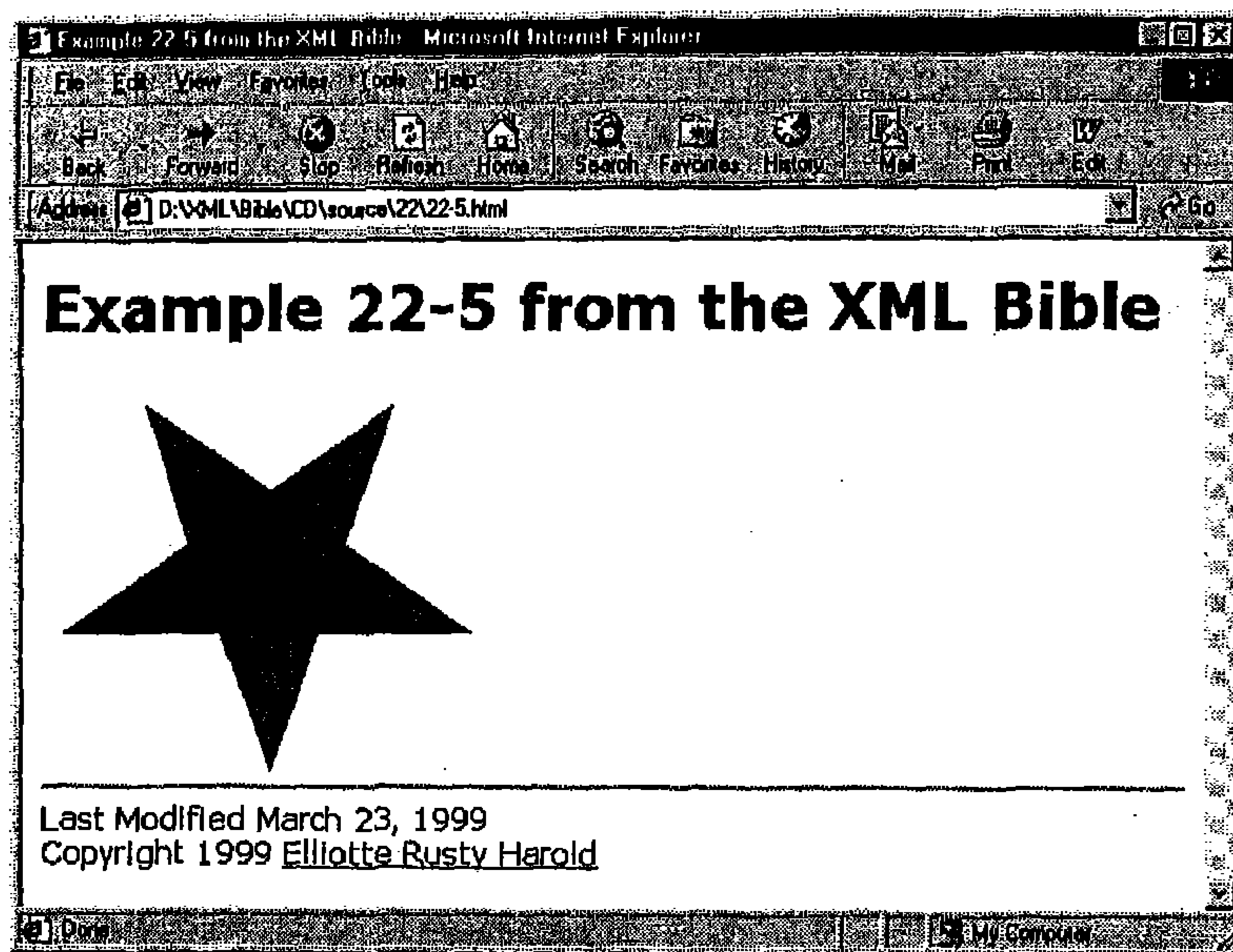
 <body>
  <h1>Example 22-4 from the XML Bible</h1>

  <div>

   <vml:polyline
    style="width: 250px; height: 250px; rotation: 120deg"
    stroke="true"
    strokecolor="black"
    strokeweight="5"
    fill="true"
    fillcolor="blue"
    points="8pt, 65pt, 72pt, 65pt, 92pt,11pt, 112pt, 65pt,
         174pt, 65pt, 122pt,100pt, 142pt, 155pt, 92pt,
         121pt, 42pt, 155pt, 60pt, 100pt, 8pt, 65pt">
   </vml:polyline>
  </div>
  <hr></hr>
  Last Modified March 23, 1999<br />
```

```
  Copyright 1999
  <a href="http://www.macfaq.com/personal.html">
   Elliotte Rusty Harold
  </a>
 </body>

</html>
```



الشكل ٢٢-٥ نجمة تدور حول محورها الســـيني

تعليمات البرمجة ٢٢-٥: نجمة تدور حول محورها السيني

```
<html xmlns:vml="urn:schemas-microsoft-com:vml">

 <head>
  <title>Example 22-5 from the XML Bible</title>
  <object id="VMLRender"
   classid="CLSID:10072CEC-8CC1-11D1-986E-00A0C955B42E">
  </object>
  <style>
```

```
    vml\:* { behavior: url(#VMLRender) }
   </style>
  </head>
 <body>
   <h1>Example 22-5 from the XML Bible</h1>
 <div>
 <vml:polyline
      style="width: 250px; height: 250px; flip: y"
      stroke="true"
      strokecolor="black"
      strokeweight="5"
      fill="true"
      fillcolor="blue"
      points="8pt, 65pt, 72pt, 65pt, 92pt,11pt, 112pt, 65pt,
             174pt, 65pt, 122pt,100pt, 142pt, 155pt, 92pt,
             121pt, 42pt, 155pt, 60pt, 100pt, 8pt, 65pt">
    </vml:polyline>
   </div>
   <hr></hr>
   Last Modified March 23, 1999<br />
   Copyright 1999
   <a href="http://www.macfaq.com/personal.html">
    Elliotte Rusty Harold
   </a>
  </body>
 </html>
```

خصائص مركز x ومركز y

تحديد خصائص center-x وcenter-y مكان مركز صندوق الكتلة الذي يحتوي على الشـــكل، كما أنها تعطي بدائل لخصائص left وright والتي توضح نفس المعلومات، وبمـــا أنـــه يمكـــن استخدام خاصتي center-x وleft بالتبادل، كذلك الحال بالنسبة لخـــاصتي center-y وright، فلا ينبغي عليك تحديد هما الاثنين وإذا قمت بتوظيف الاثنين معاً، فتستخدم القيمة التابعة لخاصتي center-x and center-y.

VML في أوفيس ٢٠٠٠

يدعم كل من تلك البرامج واكسل وباوربوينت VML، وذلك بتحول الرسوم البيانية الموجودة فـــي تلك البرامج إلى ترميز VML على صفحات HTML. وللقيام بذلك يجب إعداد منتجـــات أوفيــس بطريقة سليمة.

الإعدادات

توجد الإعدادات في نفس المكان في كل محتويات أوفيس التي يمكنها إنشاء VML. ولتعبين VML كرسوم بيانية افتراضية النوع، عليك بالخطوات التالية:

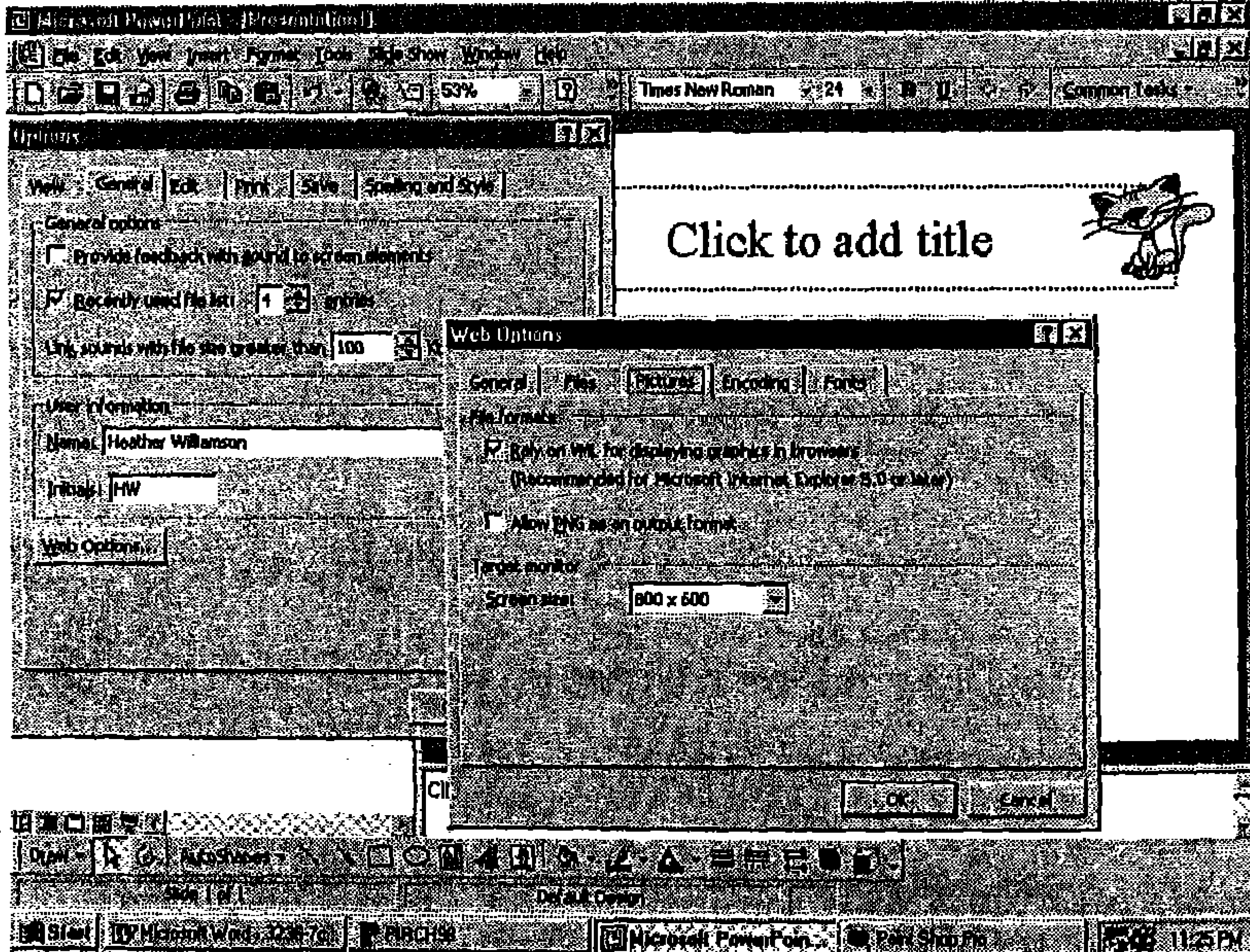
١-من مايكروسوفت باوريونيت أوورد\اكسل، افتح قائمة الأدوات الجدولة العامة.

٢-حدد الجدولة العامة.

٣-انقر زر خيارات ويب.

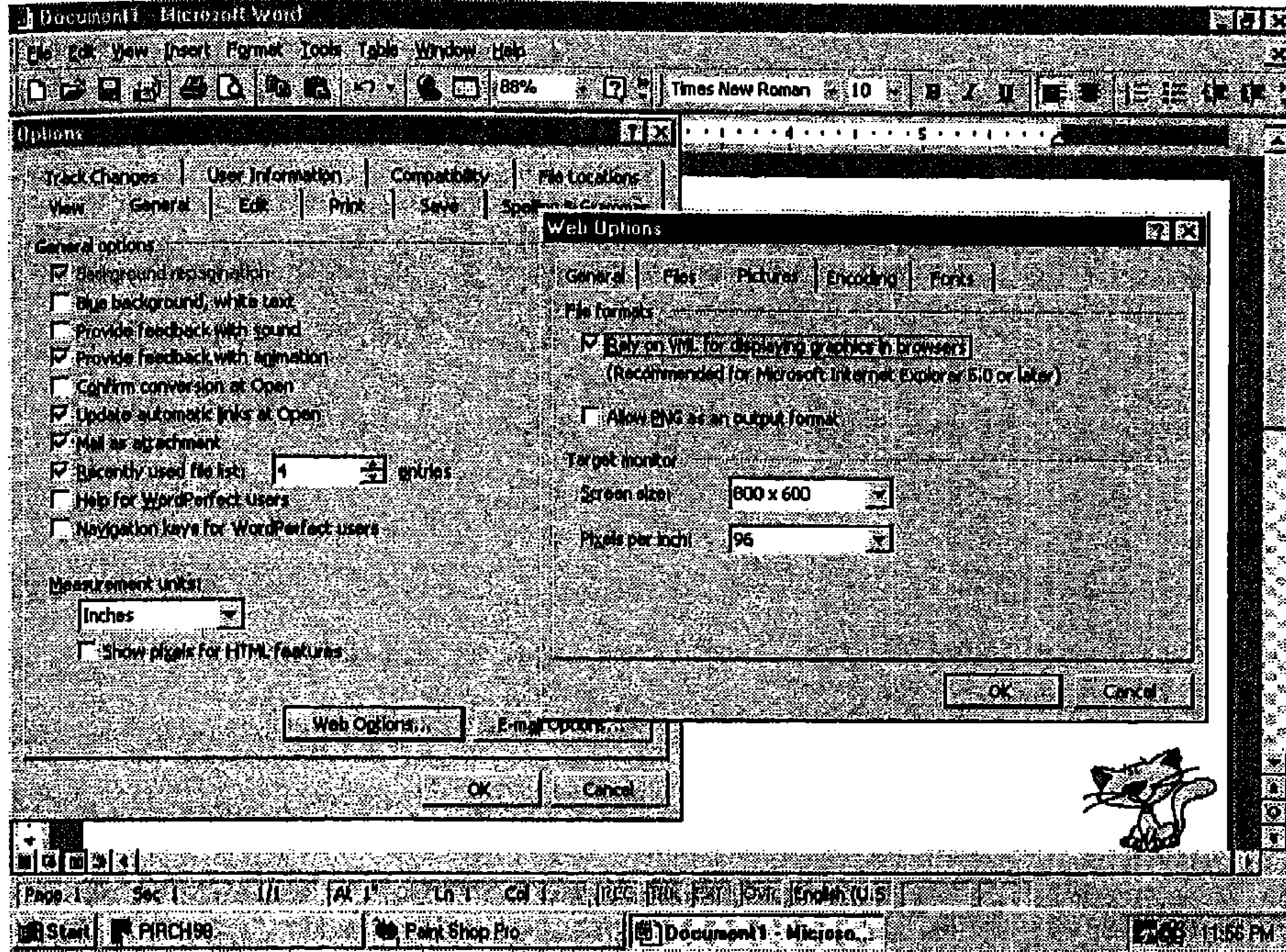
٤-حدد جدولة الصور من نافذة حوار خيارات ويب.

٥-اكتشف عن الخيار الذي يقول يعتمد على VML لعرض الرسومات في المستعرض كما هـــو موضح في الأشكال٢٢-٦ باور بوينت و٢٢-٧ ورد و٢٢-٨ إكسل.



الشكل ٢٢-٦ تعين VML كرسم افتراضي النوع في بـــاور بوينـــت.

٦-انقر زر OK في نافذة خيارات ويب أول مرة ثم في نافذة الخيارات في البرنامج الرئيسـي بعد ذلك كما هو موضح في الأشكال ٢٢-٦ و٢٢-٧ و٢٢-٨. ولأن تكون باوربوينت اوورد \الكسل لاستخدام رسومات VML كلما قمت بحفظ عرض تقديمي في تنسيق ويب.



الشكل ٢٢-٧ تعين VML كرسم افتراضي النوع فـــي مايكروســوفت ورد.

وسيصدر أوفيس ٢٠٠٠ إلى VML فقط تلك الصور التي رسمتها في المستندات الخاصة بـها مستخدمها أيضاً أدوات الرسم الخاصة بها، وهذا يعني انك لا تستطيع استخدام بـاور بوينـت أو وورد كأداة تحويلية مساعدة لرسومات أخرى قمت بتضمينها في مستندات أوفيس.

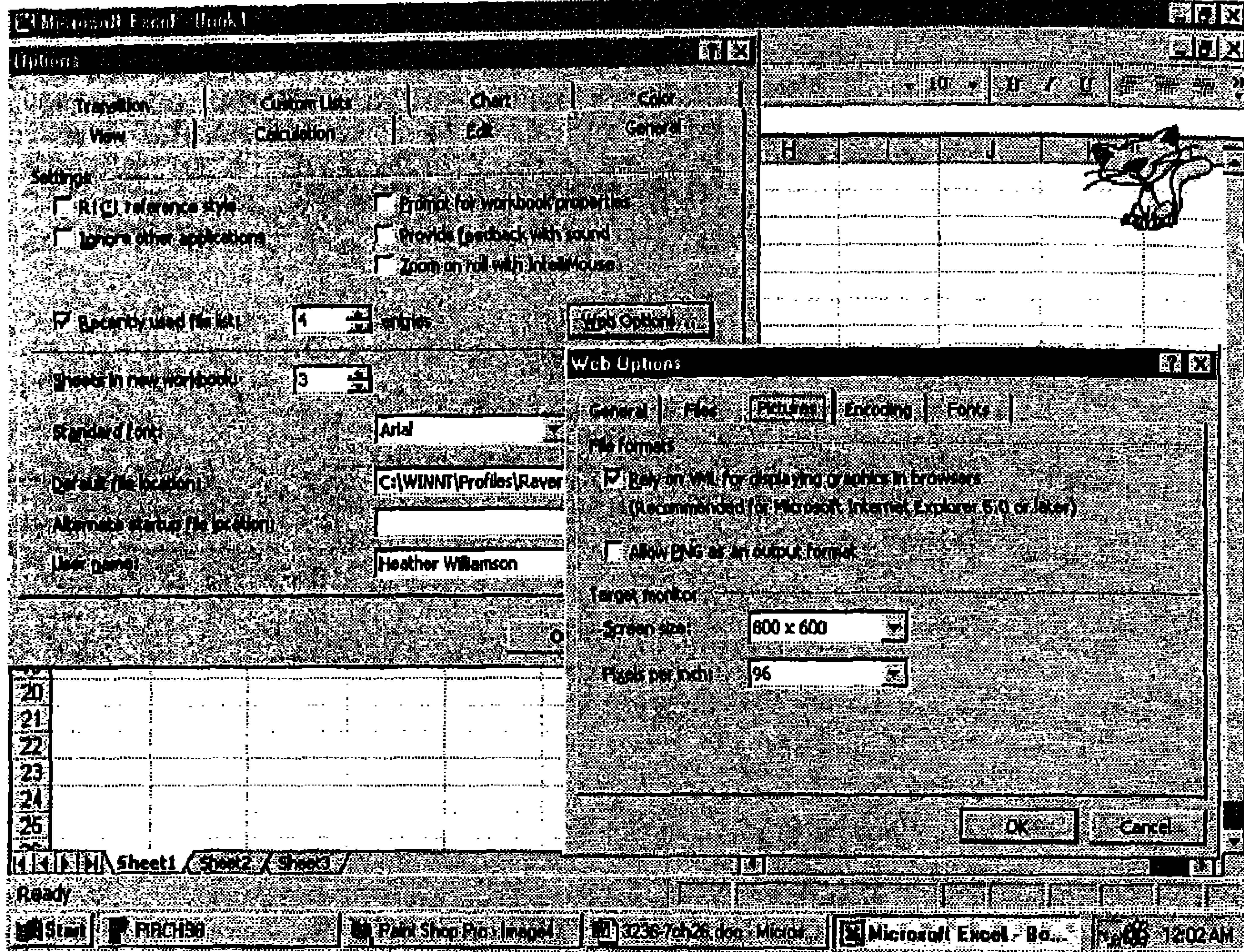
عرض مبسط لرسومات منزل

قد لا يكون أوفيس ٢٠٠٠ بنفس كفاءة وقوة Corel Draw Adobe Illustrator، لكنـة يجعـل رسم الصور البسيطة أمراً سهلاً، أسهل بكثير من الرسم باستخدام لوحة المفاتيح كما ذكر ســلفا. وممن كل محتويات أوفيس، نجد أن برنامج باور بوينت هو أكثر البرامج اهتماماً برسم الصـور، والآن سنعرض كيفية رسم منزل صغير باستخدام باور بوينت. وتطبيق تلك الخطوات ســتجدان رسم هذا المنزل سيكون سهلاً للغاية، تماماً كرسم بضعة مربعات أو دوائر أو مثلثات:

١ - افتح عرض تقديمي فارغ جديد من باور بوينت مستخدماً قائمة الملفات، الخيار الجديد.

٢- حدد عرض تقديمي فارغ ثم انقر بالفأرة على OK.

٣- ومن نافذة New Slide، حدد الشريحة التي لها شريط عنوان بأعلى، كما هو موضح في شكل ٢٢-٩، ثم انقر بالفأرة علي OK.



الشكل ٢٢-٨ تعين VML كرسم افتراضي النوع في مايكروسوفت اكسل.

٤- انقر بالفأرة ناحية شريط العنوان ثم أعط الشريحة اسماً، لنقل مثلاً My VML House.

٥- انقر بالفأرة على أداة مستطيل الموجودة في شريط أدوات الرسم ثم استخدام هذه الأداة لرسم الأساس VML House.

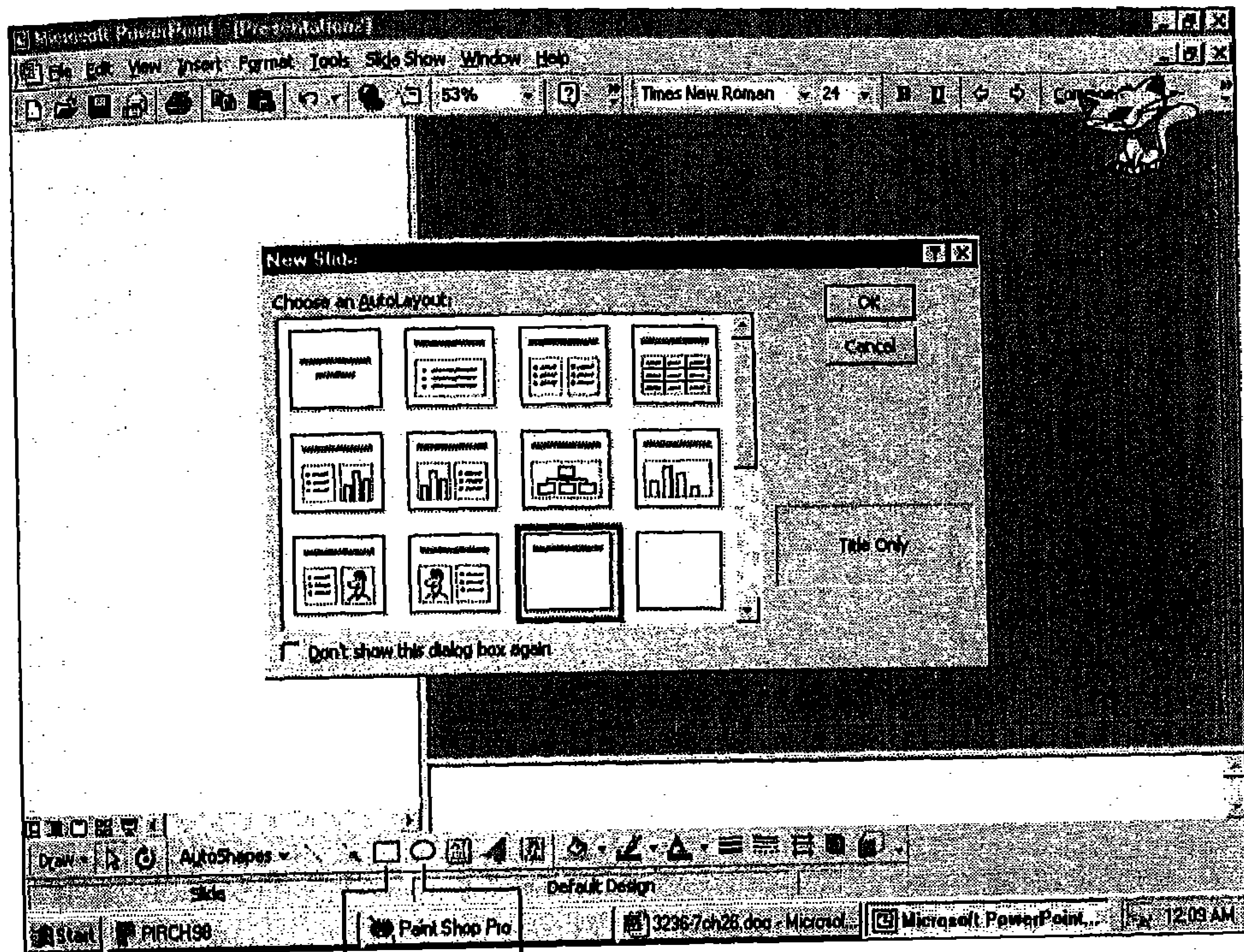
٦- وفي شريط أدوات الرسم أيضا، انقر بالفأرة على زر الأشكال التلقائية، وحدد خيار الأشكال الأساسية ثم المثلث متساوي الساقين من بعدهم.

٧- ارسم سقفاً أعلي المنزل.

٨- استخدام الأدوات البيضاوية والمستطيلة لرسم نوافذ وأبواب المنزل إلى إن تبدو الصورة التي ترسمها شبيهة تبلك الموضحة في الشكل ٢٢-١٠.

٩- افتح ملف القائمة، واختر أن تحفظ كصفحة ويب "Save As Web Page" حدد نوع اسم الصفحة، كان تقول مثلاً VMLHouse.html ثم انقر على زر Save.

١٠- اغلق باور بوينت، ثم افتح الملف الذي إنشائه مستخدما Internet Explorer 5.0، أو حدد صفحة ويب معاينة من قائمة الملف.



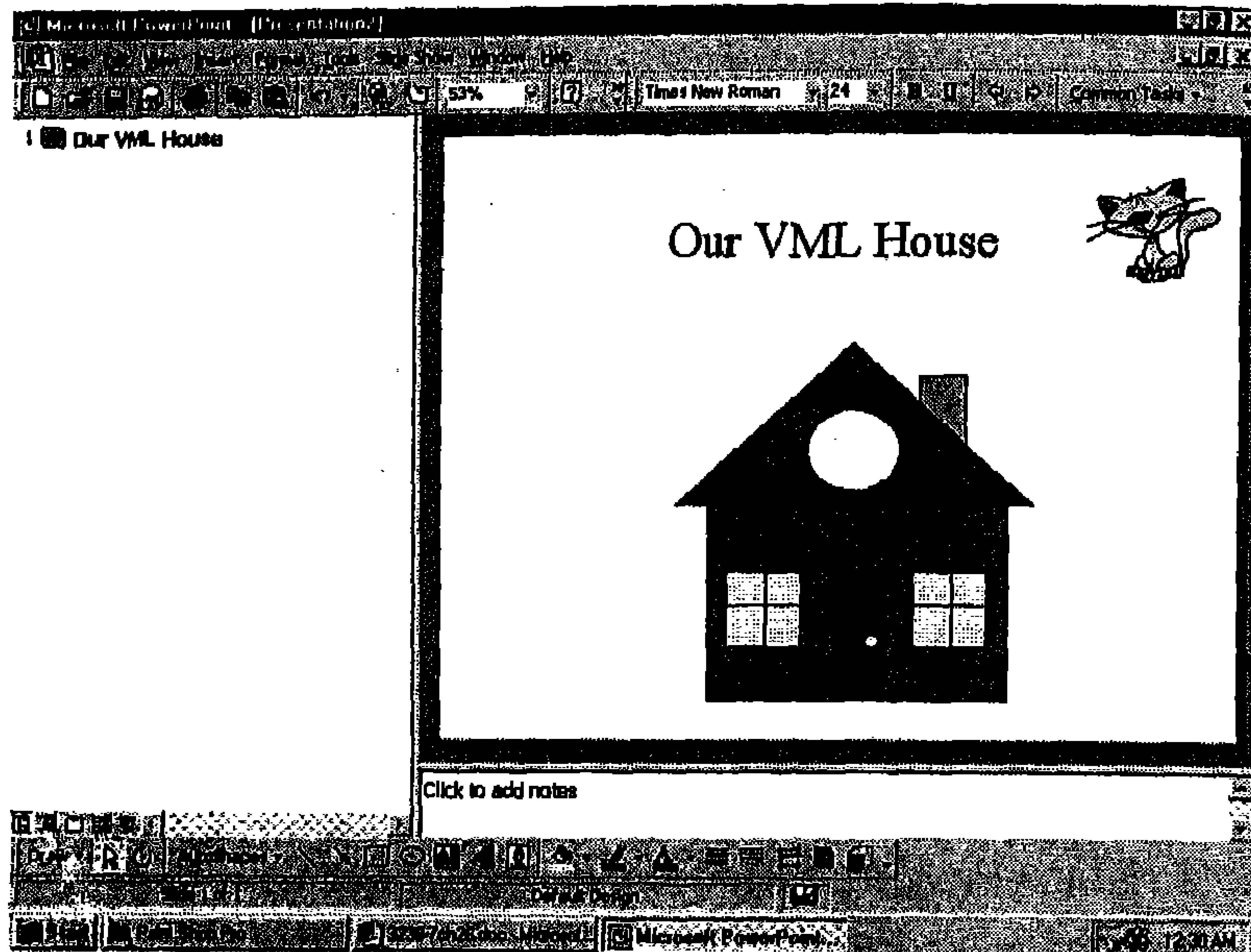
Rectangle tool   Oval tool

الشكل ٢٢-٩ تحديد قالب لشـــريحة.

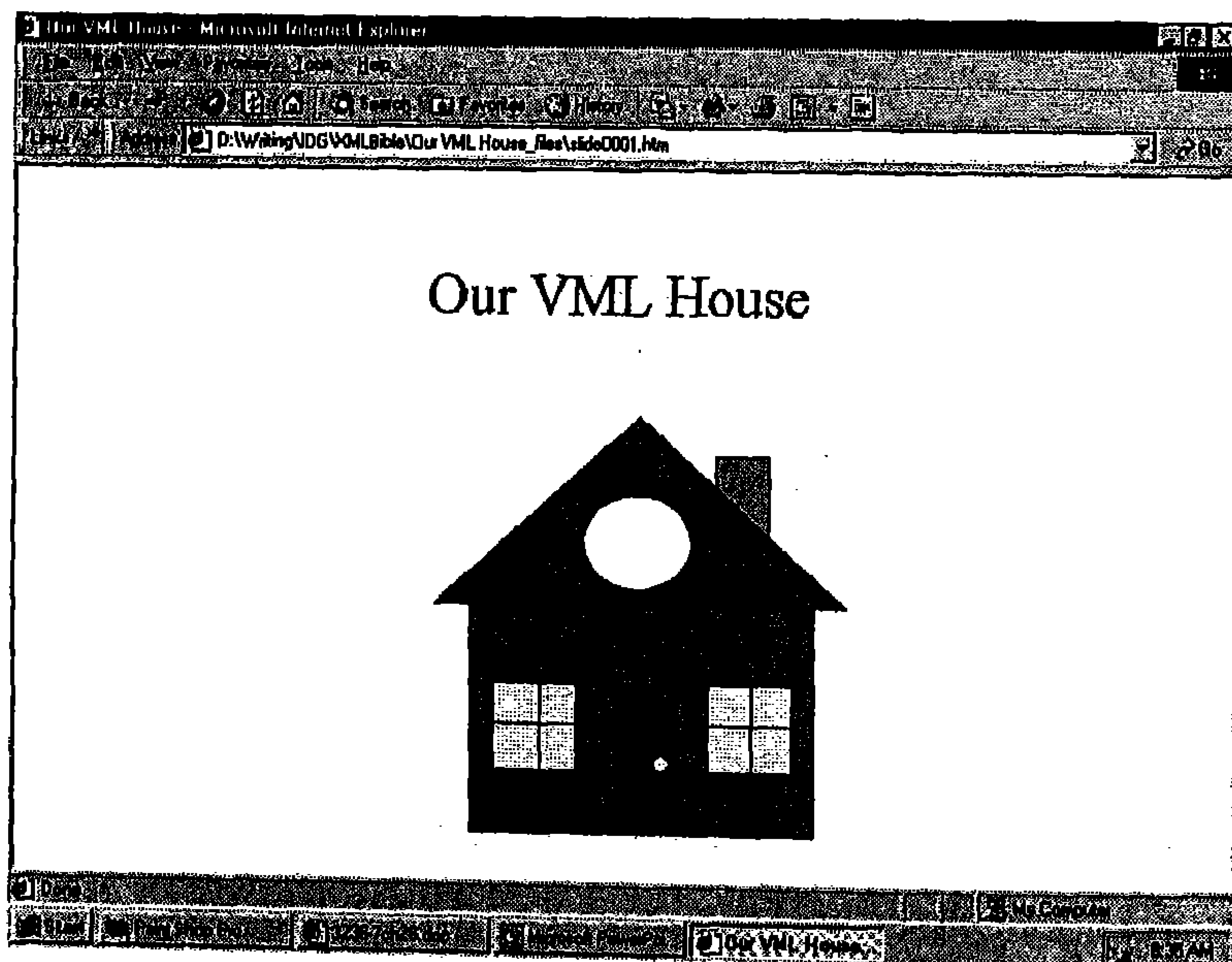
يوضح الشكل ٢٢-١١ الشكل النهائي لصفحة ويب. أما عن تعليمات HTMLوVMLالبرمجـة التي إنشائها باور يونيت لتستخدم لعرض تعليمات HTML وVML البرمجة القياسية، هناك أيضـاً عدد لا بأس به من العنـــاصر فـي urn:schemas-microsoft-com:office:office وفـي مسافات الأسماء urn:schemas-microsoft-com:office: powerpoint وجميعها تحتـوي على معلومات لن يستخدمها معظم مستعرض ويب لكن سيستخدمها باور يوينت لو تم فتح ملـف HTML فيه. وتهدف هذه العناصر إلى السماح للمستند بالقيام برحلة مستديرة تبدأ من باور بوينت ثم إلى HTML ثم إلى باور بوينت مرة أخرى دون أن تفقد شيئاً أثناء تلك الرحلة.



سيظهر المنزل VML فقط في Internet Explorer 5.0، أما عن مســتعرضي Netscape فلا يمكنها سوى رؤية الصور المضمنة.



الشكل ٢٢-١٠ VML House في باور بوينت ٢٠٠٠ بعد أن أصبح جاهزاً لتحويله إلى نص VML.



الشكل ٢٢-١١ عرض VML House كصفحة ويب في Internet Explorer 5.0.

تعليمات البرمجة ٢٢-٦: تحويل شريحة باور بوينت Our VML House إلى ملف HTML متضمنا VML للاستخدام على ويب

```
<html xmlns:v="urn:schemas-microsoft-com:vml"
    xmlns:o="urn:schemas-microsoft-com:office:office"
    xmlns:p="urn:schemas-microsoft-com:office:powerpoint"
    xmlns="-//W3C//DTD HTML 4.0//EN">

<head>
  <meta http-equiv=Content-Type content="text/html;
     charset=windows-1252">
  <meta name=ProgId content=PowerPoint.Slide>
  <meta name=Generator content="Microsoft PowerPoint 9">
  <link id=Main-File rel=Main-File
     href="../Our%20VML%20House.htm">
  <link rel=Preview href=preview.wmf>

  <!-[if !mso]>
   <style>
    v\:* {behavior:url(#default#VML);}
    o\:* {behavior:url(#default#VML);}
    p\:* {behavior:url(#default#VML);}
    shape {behavior:url(#default#VML);}
    v\:textbox {display:none;}
   </style>
  <![endif]>
  <title>Our VML House</title>
  <meta name=Description content="8-Mar-99: Our VML House">
  <link rel=Stylesheet href="master03_stylesheet.css">

  <![if !ppt]>
   <style media=print>
   <!-.sld
```



```
      {left:0px !important;
      width:6.0in !important;
      height:4.5in !important;
      font-size:103% !important;}
    ->
    </style>
    <script src=script.js>
    </script>
    <!-[if vml]>
      <script>
       g_vml = 1;
      </script>
    <![endif]>
    <script for=window event=onload>
      <!-LoadSld( gId );
      MakeSldVis(0);
      //->
    </script>
  <![endif]>
  <o:shapelayout v:ext="edit">
    <o:idmap v:ext="edit" data="2"/>
  </o:shapelayout>
</head>
<body lang=EN-US style='margin:0px;background-color:white'
     onresize="_RSW()">
  <div id=SlideObj class=sld
      style='position:absolute;top:0px;left:0px;
            width:554px;height:415px;font-size:16px;
            background-color:white;clip:
            rect(0%, 101%, 101%, 0%);
            visibility:hidden'>
  <p:slide coordsize="720'540"
   colors="#FFFFFF,#000000,#808080,#000000,#00CC99,#3333CC,
```

```
        #CCCCFF,#B2B2B2"
  masterhref="master03.xml">
<p:shaperange href="master03.xml#_x0000_s1025"/>
<![if !ppt]>
  <p:shaperange href="master03.xml#_x0000_s1028"/>
  <![if !vml]>
  <img border=0
      v:shapes="_x0000_s1028"
      src="master03_image002.gif"
      style='position:absolute;top:91.08%;left:7.58%;
            width:21.11%;height:6.98%'>
  <![endif]>
  <p:shaperange href="master03.xml#_x0000_s1029"/>
  <![if !vml]>
    <img border=0
      v:shapes="_x0000_s1029"
      src="master03_image003.gif"
      style='position:absolute;top:91.08%;left:34.11%;
            width:31.94%;height:6.98%'>
  <![endif]>
<![endif]>
<v:rect id="_x0000_s2063"
      style='position:absolute;left:438pt;top:3in;
            width:42pt; height:78pt;mso-wrap-style:
            none;v-text-anchor:middle'
      fillcolor="#0c9 [4]"
      strokecolor="black [1]">
  <v:fill color2="white [0]"/>
  <v:shadow color="gray [2]"/>
</v:rect>
<p:shaperange href="master03.xml#_x0000_m1026"/>
<v:shape id="_x0000_s2050"
      type="#_x0000_m1026"
```



```
        style='position:absolute;left:54pt;top:48pt;
              width:612pt; height:90pt'>
    <v:fill o:detectmouseclick="f"/>
    <v:stroke o:forcedash="f"/>
<o:lock v:ext="edit" text="f"/>
    <p:placeholder type="title"/>
  </v:shape>
  <v:rect id="_x0000_s2051"
        style='position:absolute; left:246pt;top:330pt;
              width:270pt;height:174pt;mso-wrap-style:none;
              v-text-anchor:middle'
        fillcolor="red"
        strokecolor="black [1]">
    <v:shadow color="gray [2]"/>
  </v:rect>
  <v:shapetype id="_x0000_t5"
            coordsize="216000,21600"
            o:spt="5"
            adj="10800"
            path="m@0,0l0,21600,21600,21600xe">
    <v:stroke joinstyle="miter"/>
    <v:formulas>
      <v:f eqn="val #0"/>
      <v:f eqn="prod #0 1 2"/>
      <v:f eqn="sum @1 10800 0"/>
    </v:formulas>
    <v:path gradientshapeok="t"
          o:connecttype="custom"
          o:connectlocs="@0,0;@1,10800;0,21600;10800,21600;
                    21600,21600;@2,10800"
          textboxrect="0,10800,10800,18000;
                    5400,10800,16200,18000;
```

```
                    10800,10800,21600,18000;
                    0,7200,7200,21600;
                    7200,7200,14400,21600;
                    14400,7200,21600,21600"/>
  <v:handles>
   <v:h position="#0,topLeft" xrange="0'21600"/>
  </v:handles>
</v:shapetype>
<v:shape id="_x0000_s2053"
       type="#_x0000_t5"
       style='position:absolute;left:3in;top:186pt;
            width:324pt;height:2in;mso-wrap-style:none;
            v-text-anchor:middle'
       fillcolor="#33c [5]"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:shape>
<v:oval id="_x0000_s2054"
      style='position:absolute;left:336pt;top:246pt;
      width:84pt;height:1in;mso-wrap-style:none;
           v-text-anchor:middle'
      fillcolor="white [0]"
      strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:oval>
<v:rect id="_x0000_s2055"
      style='position:absolute;left:264pt;top:390pt;
           width:66pt;height:66pt;mso-wrap-style:none;
           v-text-anchor:middle'
      fillcolor="#6ff"
      strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
```

```
</v:rect>
<v:rect id="_x0000_s2056"
       style='position:absolute;left:5in;top:390pt;
              width:48pt;height:114pt;mso-wrap-style:none;
              v-text-anchor:middle'
       fillcolor="black [1]"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:rect>
<v:rect id="_x0000_s2057"
       style='position:absolute;left:6in;top:390pt;
              width:66pt;height:66pt;mso-wrap-style:none;
              v-text-anchor:middle'
       fillcolor="#6ff"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:rect>
<v:line id="_x0000_s2058"
       style='position:absolute'
       from="300pt,390pt"
       to="300pt,456pt"
       coordsize="21600,21600"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:line>
<v:line id="_x0000_s2059"
       style='position:absolute'
       from="264pt,420pt"
       to="330pt,420pt"
       coordsize="21600,21600"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
```

```
</v:line>
<v:line id="_x0000_s2060"
      style='position:absolute'
      from="468pt,390pt"
      to="468pt,456pt"
      coordsize="21600, 21600"
      strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
 </v:line>
 <v:line id="_x0000_s2061"
       style='position:absolute'
from="6in,420pt"
       to="498pt,420pt"
       coordsize="21600,21600"
       strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:line>
<v:oval id="_x0000_s2062"
      style='position:absolute;left:390pt;top:444pt;
           width:12pt;height:12pt;mso-wrap-style:none;
           v-text-anchor:middle'
      fillcolor="yellow"
      strokecolor="black [1]">
  <v:shadow color="gray [2]"/>
</v:oval>
<![if !vml]>
  <img border=0
     v:shapes="_x0000_s2063,_x0000_s2050,_x0000_s2051,
            _x0000_s2053,_x0000_s2054,_x0000_s2055,
            _x0000_s2056,_x0000_s2057,_x0000_s2058,
            _x0000_s2059,_x0000_s2060,_x0000_s2061,
            _x0000_s2062"
     src="slide0001_image004.gif"
```

```
        style='position:absolute;top:8.91%;left:7.58%;
            width:85.37%;height:84.81%'>
    <![endif]>
    <div v:shape="_x0000_s2050" class=T
       style='position:absolute;top:13.01%;
       left:8.48%;width:83.21%;height:9.15%'>
     Our VML House
    </div>
    </p:slide>
    </div>
  </body>
  </html>
```

نظرة سريعة على SVG

وللأمانة العلمية، يجب أن نعلمكم بان VML قد يكون تقنية أخرى لمايكروسوفت، يسير على نفس خطى ActiveX وBob، أي إنها تقنية لا يمكن تنفيذها إلا بواسطة مايكروسوفت. وNetscape Navigator لا يدعم VMLولا يبدو أنة سيدعمه يوماً ما.

لقد تلقى W3C أربعة اقتراحات مختلفة لمجموعة متنوعة من البائعين بشأن الرسوم الاتجاهية في XML. وقد شكلت هذه الاقتراحات المجموعات العاملة للرسوم الاتجاهية القابلة لتغيير الأبعاد (SVG) والتي تتكون من مثلين من هؤلاء البائعين لتطوير مواصفة واحدة لتمثيل XML للرسم الاتجاهية القابلة لتغير الأبعاد. وعندما يكتمل SVG يكون بذلك قادراً على أن يمد المستخدم بكل ما يمده به VML ويزيد على ذلك الكثير مثل إظهار الحركة، والعناصر التبادلية، والتصفيات، والآلنقاط، والتأليف والتعبئة بالنقش غير أن مواصفة SVG الكاملة والبرامج الذي ينفذ المواصفة قد أصبحا غائبين بعض الشيء، بينما أصبح VML هو الموجود اليوم.

وقد حرر الاتحاد المالي للويب أول مسودة عمل SVG، وذلك في فبراير ١٩٩٩ وقام بمراجعتها في شهر أبريل من نفس العام. وتعد هذه المسودة غير كاملة بالمقارنة بمسودات العمل الأخرى، فهي لا تتعدى أكثر من كونها تحديداً لعناصر عن كيفية أحاله تلك العناصر إلى تعليمات برمجة في XML لكني لن أكون مندهشاً لو تم إسناد الزعامة لتلك المسودة لبني المجهودات المتنافسة، التي من أهمها VML.

ولقد أعلنت مايكروسوفت على الملء انه سيتم تجاهل أي مجهودات تبذل في مجال رسومات ويب فيما عدا VML، وعلى الرغم من ذلك فإن كل هذه المجهودات توضحها مجموعة عمل

SVG وسواء كان ممثلو SVG يشاركون بالفعل أو كان اسم مايكروسوفت مجرد اسم في البيانات الإدارية، ففي كلتا الحالتين يعتبر SVG تطويرا سنشاهد في دورات التطوير القليلة التالية لنعرف إلى أين يتجه ؟ ومن الذي يتابعه ومن الذي يقوده.

خلاصة

تعرفنا في هذا الفصل على تطوير رسومات VML لاستخدامها مع Internet Explorer، فإلى جانب النظرة العامة الشاملة على VML، تعرفنا تحديدا على النقاط التالية:

- ما الذي يمكن أن تحققه VML لرسومات ويب.
- العناصر والسمات المتعددة المرتبطة بأشكال VML، وكيفية استخدامها لإنشاء ما تحتاج من صور مرئية.
- كيفية تكوين تطبيقات مايكروسوفت أوفيس ٢٠٠٠ تستخدم VML عند إنشاء رسومات للعرض التقدمية ولمستندات ويب.
- كيفية رسم أشكال VML باستخدام باور بوينت ٢٠٠٠.
- مدى تأثير SVG على رسومات ويب، ونتيجة ذلك على VML.

تناولت الفصول القليلة الأخيرة، بما فيها هذا الفصل، مجموعة متنوعة من تطبيقات XML صممتها ثلاث مجموعات وفي الفصل البداية حتى يتناول علم الأنساب.

٢٣

تصميم تطبيق XML جديد

تناولت الفصول العدة الأخيرة تطبيقات XML التي كتبها أفراد آخرون كما تناولت أيضاً كيفية استخدامها. وفي هذا الفصل تحديداً ستشاهد التطور التدريجي من نقطة البداية لتطبيق XML وDTDs مقترنة ببيانات نسبية.

تنظيم البيانات

عند تطوير تطبيق XML، يجب أن تقوم بتنظيم البيانات التي تصفها، إما في رأسك أو على الورق، وذلك من خلال ثلاث خطوات أساسية:

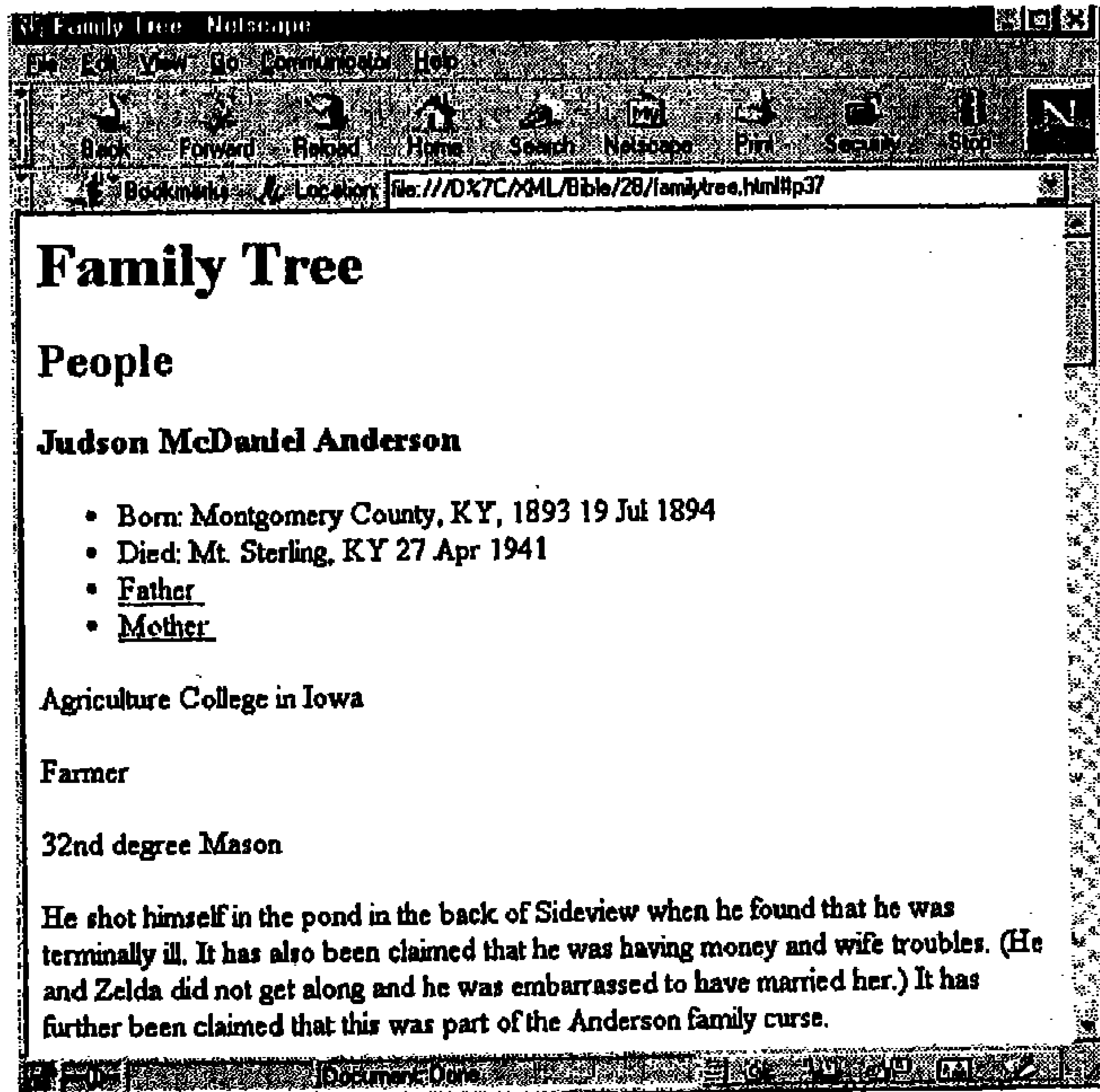
١- ضع قائمة بالعناصر

٢- حدد العناصر الرئيسية

٣- اربط العناصر بعضها ببعض

وإحدى أسهل الطرق لبدء تنفيذ عملية التنظيم تلك هي استكشاف الأشكال والتقارير المتوفرة بالفعل من تنسيقات أخرى تصف البيانات ويعد علم الأنساب علماً ذا أساس متين، ويتميز الاختصاصيين بهذا العلم بقدرتهم على تميز المفيد من المعلومات عن غيرها، وينسقها، وعادة ما يكون ذلك موجوداً في ورقة تجميع العائلة، والذي يوضح الشكل ٢٣-١ نموذجاً منه.

ستحتاج إلى نسخ الحقول من التقارير القياسية إلى DTD وتنظيمها بطريقة ما، حتى أنه يمكنك الإضافية عليها أو العديل بها بحيث تناسب احتياجات المحددة.



الشكل ٢٣-١ ورقة لمجموعة عائلية.



سيلحظ المبرمجين المهتمون بالكائن وجود العديد من نقاط التشابه بين ما جاء في هذا القسم وبين ما جمعوه من تقنيات كمتطلبات للمستخدم لعل ذلك يرجع إلى خبراتي وأدائي كمبرمج مهتم بالكائن لكنة يرجع في الأغلب إلى التشابه بين المهام المتناولة. ولا يختلف جمع متطلبات المستخدم للبرنامج كثيراً عن جمع متطلبات المستخدم للغات. وقد يلحظ أيضاً مصممو قاعدة البيانات العديد من أوجه الشبة بين ما يقدم هنا وبين ما يعتمدون عليه عند صميم قاعدة بيانات جديدة.

وضع قائمة بالعناصر

أول خطوة في تطوير DTD لمجال ما هي تحديد العناصر، وهذه ليست بالخطوة الصعبة فهي عادة ما يتم عن طريق إبداع فكرة بارعة مفاجئة لتحديد ما الذي يمكن أن يحتوي علية المجال. وكنوع من التمرين، اكتب كل ما يخطر ببالك ويمكن إن يقدم معلومات عن علم الأنساب، وقم يتضمن أية بيانات عن هذا العلم حتى يسهل عليك التغلب على المشكلة. افترض مثلاً أنة بإمكانك استخدام XHTML DTD التي ورد ذكرهما في الفصل العشرين لمعلومات النص القياسية مثل الفقرات وعناوين الصفحات، لكن مرة أخرى ننصح بتضمين العناصر فقط التي تنطبق على علم الأنساب.

لا تخجل إذا وقعت في خطأ ما فإنه يمكنك فيما بعد التخليص من المعلومات الزائدة أو تلك التي لا تثبت فائتها، ومن ثم توقع في هذه المرحلة أن يكون هناك عناصر مكررة، تتخلص منها بعد التفكير العميق.

وتلك هي القائمة التي تواصلت إلى وضعها، وربما القائمة التي ستقوم أنت بوضعها ستكون مختلفة بعض الشيء، فقد تستخدم مسميات مختلفة لكن كلها تشير إلى نفس الأشياء، وذلك لأبهتم، فلا يمكن الاتفاق على أجابه واحدة فقط (غير أن هذا لا يعني أن جميع الإجابات متساوية أو أنة لا توجد إجابات أفضل من غيرها)

الأب	العائلة
الأب والأم	عيد ميلاد
المعمودية	دفن
الملاحظة	اللقب
العمة أو خالة	الجدة
الأم	الابن
الابن	تاريخ الوفاة
التبني	الجد
مكان القبر	اسم التنصير
ابنة الأخ أو الأخت	العم

الفرد
الطفل "رضيع"
النوع
المصدر
الجد أو الجدة
الابنة
الزواج
التاريخ
الاسم الأوسط
ابن الأخ أو الأخت

تحديد العناصر الرئيسية

تقسم القائمة التي جاءت في القسم الخير بان لها نسخاً رائعة ولها أيضاً بعض العناصر غير الهامة، وإذا ما تماديت ستجد أنها تفتقد أيضاً بضعة عناصر، ولا يعد لك غريباً بل طبيعي جداً، فتطوير تطبيق XML تعد عملية مكررة لا تشعر بالرضاعة عن نتائجها إلا بع فترة من الوقت.

وتحديد العناصر الرئيسية للمجال هو كل ما عليك القيام به في تلك المرحلة، وهذه العناصر عادة ما تكون تلك التي تحتوي عليها بعض العناصر الأخرى. أما عن الإمكانيات الحقيقية هنا فهي اثنتان: العائلة الأخرى. أما عن الإمكانيات الحقيقية هنا فهي اثنتان: العائلة والفرد، بينما تعد العناصر الأخرى الموجودة بالقائمة إما العائلة والفرد والعائلة "محل العمل أو الإقامة عيد الميلاد، والزواج" أو نوع لهما "العم، الأب والأم، والطفل الرضيع".

في هذه المرحلة، نجد أن الجميع يميلون إلى القول بأن العائلة هي العنصر الرئيسي الوحيد، وأن عناصر العائلات تحتوي بدورها على الأفراد، وهذا ما ينسق بالفعل مع استخدام المصطلحات parent وchild الحديث عن العلاقات بين عناصر XML "وهو ما تم استخدامه في هذا الفصل حتى لا يختلط عليك الأمر عن الأباء والأبناء الذين تخذهم نموذجاً" فعلى سبيل المثال يمكن أن تكون العائلة على النحو عند الحديث التالي:

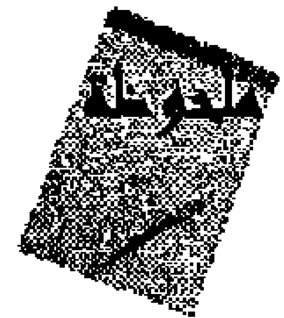
```
<FAMILY>
  <HUSBAND>Samuel English Anderson</HUSBAND>
  <WIFE>Cora Rucker McDaniel</WIFE>
  <CHILD>Judson McDaniel Anderson</CHILD>
  <CHILD>Thomas Corwin Anderson</CHILD>
  <CHILD>Rodger French Anderson</CHILD>
  <CHILD>Mary English Anderson</CHILD>
</FAMILY>
```

غير أن هذه الطريقة قد تواجهها مشكلة ما، فالفرد الواحد ينتمي إلى أكثر من عائلة، فعلى سبيل المثال أنا ابن لأب وأم وفي نفس الوقت زوج لامرأة أخرى من عائلة أخرى، إذن يعتبرون كلتا الأسرتان أسرة واحدة ممتدة، لكن لو نظرنا إلى جذور تلك العائلة فقط من أب وأم وأبناء وذلك من أجل تدوين البيانات. لكن النماذج الموجودة في واقع الحياة ليست بهذه البساطة، فنجدة لبعض الأفراد أباء وأمهات إما عن طريق التبني أو باستخدام الطرق البيولوجية، إلى جانب

هؤلاء الذين يتزوجون أكثر من مرة، فوالد زوجتي سيدني هارت أتدرسون مثلاً قد تزوج خمسة عشر مرة من اثنتي عشر امرأة، لكنة يعد حالة مبالغ فيها، حتى وفاته قد حرمته من الزواج أربعة مرات أخرى ومن الوصول إلى الرقم القياسي العالمي المسجل لعدد الزيجات المتتالية "ومنذ ذلك الحين اصدر المعمد والوزير السابق جلابين وولف قراراً برفع الرقم القياسي للزيجات المتتالية ليصبح ثمانية وعشرون" غير أنة ليس عليك أن تضع في الاعتبار هؤلاء الأفراد ممن ينتمون إلي عائلات مختلفة.

ويفسر هذه الظاهرة أوراق تجميع العائلة القياسية التي يستخدمها مورمونز. والتي رأينا إحدى أشكالها في الشكل ٢٣-١، وذلك عن طريق تكرار نفس الأفراد ونفس البيانات على أوراق مختلفة لكن بالنسبة لتطبيقات الكمبيوتر، فإن يفضل عدم تكرار نفس المعلومات أكثر من مرة وبالتالي يجتنبا ذلك المشكلات الخاصة بتحديث بعض البيانات المخزنة بمكان ما، وترك البعض الأخر المخزن في مكان أخر. وبدلا من ذلك يمكنك الربط بين العناصر المختلفة باستخدام سمات ID و IDREF.

ومن ثم يتضح أن عنصر عائلة family رئيسي واحد لا يكفي، بل يجب أن يكون هناك على الأقل عنصر رئيسي أخر، وهو الفرد "person". فلكل فرد= شخصية الفريدة، فلكل عيد ميلاد واحد إلى غير ذلك من البيانات. وبما أن العائلات تتكون من مجموعات مختلفة من الأفراد، فإن تعريف هؤلاء الأفراد والأدوار التي يقومون بها داخل العائلة تمكنك من تعريف العائلة ذاتها.



عادة ما تعتقد أن الأسرة تشمل الجدود والجدات، وزوجات الأبناء الأعمام والعمات والخالات وأبناء العم وربما أفراد آخرين لا تربطنا بهم أي صلة سوى أنهم يقيمون بنفس المنزل. لكن من منظور علم الأنساب تتكون الأسرة من الأب والأم والأبناء، غير أنه وفي كثير الأحيان قد تكون أسماء هؤلاء الإفراد غير معلومة، وفي كثير من الحالات لا يكون هناك أطفال أوقد لا يكون هنلك زوج أو زوجة (فيمثل فرد واحد الأسرة). ولا تتعدى الأسرة أكثر من تلك العلاقات، لذا يتناول جزء كبير من علم الأنساب إقامة العلاقات بين الإفراد عن طريق التبني أو بالطريق البيولوجية لذلك يعد غريباً أن تكتشف في بحث أحدهم أن Cousin Puss أو Aunt Moot مثلاً الذي يشير إليهم لا تربطه بهم أي علاقة! يمكنك إدراج مثل هؤلاء الأفراد في تسجيلاتك لكن اعلم جيداً أن فشلك في إيجاد العلاقات الحقيقة لهؤلاء الأفراد سيزيد الأمور تعقيداً.

هناك عنصر أساسي أخر قد يكون نتيجة مباشرة للجذر، وهو مصدر المعلومات. يبدو المصدر وكأنه حاشية سفلية بليوغرافية تحديد من أين أخذت كل معلومة. وقد يكون المصدر مقال

من مجلة مثل 1987 و26-٣١ Blaise Pradel, Man At Arms, May/June 1987, p. أو كتاباً مثل A Sesquicentennial History of Kentucky by Frederik A. Wallis & Hambleon Tapp, 1945, The Historical Record Association, Hopkinsville, KY أو يكون مأخوذاً من الإنجيل مثل English-Demint Anderson Bible أو يكون جـــاء علـــى لسان أحد الأشخاص مثل Anne Sandusky, interview, 6-12-1995.

ويعد تبع المصدر لإحدى البيانات مهماً لأنه عادة لا تتفق المصادر المختلفة، فلا غرابة فـــي أن يكون الفرق بين تواريخ الميلاد والوفاة يوماً أو سنة قبل أو بعد. أما عن الخلط بيـــن الأبـــاء والأجداد، العمات وأبناء العم وأسماء بعض الأفراد فهو أقل شيوعاً لكنه أيضاً يوجد بكثيرة. لذلـك عندما تقوم باكتشاف معلومة جديدة تجعل المعلومة القديمة في موقع نقاش، يلزم عليك الحكم على تلك المعلومة الجديدة ما إذا كان يمكن الاعتماد عليها أكثر من المعلومة القديمة، وذلك لأن ليـــس كل المصادر يمكن الاعتماد عليها. وفي البحث الخاص بي وجدت مستنداً يرجع نسب زوجي إلى آدم وحواء ثم الأسر الإنجليزية المالكة في العصور الوسطى مستعيناً ببعض الشخصيات المأخوذة من الإنجيل. غير أنني لم آخذ هذا المصدر مأخذ الجد.

يمكنني في بعض الأسباب المعقولة التي تجعل المصدر نتيجة للعناصر الفردية التي يدعمـــها بالوثائق، لكنني أعتقد أن المصدر لا يعد جزء للفرد أو للأسرة تماماً مثلماً أرى أن تاريخ الميـلاد أو الزواج ينتمي إلى فرد بعينة، لأن المصدر ما هو إلا معلومات يجب تخزينها منفصلة والإشارة إليها من خلال رقم معرف. ولعل السبب الرئيسي لذلك هو أن المصدر الواحد قد يحتوي بـــالفعل على بيانات خاصة بعناصر مختلفة. لذا، التزاماً بمبادئ تطبيع البيانات، فأنا أفضل عدم تكـــرار المعلومات عن المصدر أكثر من مرة في المستند، وإذا رغبت في ذلك فان يكون ماثلاً لاسـتخدام التعليمات الختامية بدلاً من الحواشي السفلية.

إقامة علاقات بين العناصر أو "الربط بين العناصر"

الخطوة الثالثة والأخيرة قبل البدء في كتابة DTD هي تحديد كيفية ترابـــط المعلومـــات المـــراد تتبعها. ولقد ذكرت أن العناصر الثلاث الرئيسية هي الفرد والأسرة والمصدر، الآن عليك تقريـــر ما تم تضمينه في تلك العناصر الرئيسية الثلاث.

الأسرة FAMILY

عادة ما تتكون الأسرة من زوج وزوجة وأبناء،وقد لا يكون هناك أبناء، وقد يكون أحد الزوجيـــن غير موجود. وإذا أردت تناول الزيجات من نفس النوع "الأمر الذي لم يســـتطيع برنـــامج علـــم الأنساب القيام به حتى اليوم" فلا تحدد النوع حيث أنه سيكون ضمن سمة لفرد التي ينتمي إليـــها على أية حال.

هل توجد معلومات أخرى ترتبط بالأسرة في مقابل الأفراد؟ إنني أرى أن المعلومات عن الزواج مهمة بالنسبة للاختصاصيين في علم الأنساب، علي سبيل المثال تاريخ ومحل الزواج "إن وجد" وتاريخ ومحل الطلاق "إن وجد أيضاً". وعلى الرغم من إن هذه التواريخ تعد جزء من كل فرد متزوج، فإنها تعد أيضا جزء من الأسرة. وبهذه المعطيات قد تبدو الأسرة كالأتي.

```
<FAMILY>
  <MARRIAGE>
    <DATE>...</DATE>
    <PLACE>...</PLACE>
  </MARRIAGE>
  <DIVORCE>
    <DATE>...</DATE>
    <PLACE>...</PLACE>
  </DIVORCE>
  <HUSBAND>...</HUSBAND>
  <WIFE>...</WIFE>
  <CHILD>...</CHILD>
  <CHILD>...</CHILD>
  <CHILD>...</CHILD>
</FAMILY>
```

ويمكن حذف بعض المعلومات إن لم تكن وثيقة الصلة بالموضوع "مثل المعلومات الخاصة بالطلاق لزوجين لم يطلقا" وإن لم تكن متوفرة.

الفرد PERSON

يبدو عنصر PERSON أكثر تعقيداً ولنستعرض المعلومات القياسية التي يجب تخزينها عن الفرد:

- الاسم
- الميلاد
- المعمودية "الديانة"
- الموت
- الدفن
- الأب
- الأم

وما سبق نقول أن الاسم والميلاد والموت والدفن هي عناصر يحتوي عليها الفرد، بينما الأب والأم هما سمات للفرد تشير بدورها إلي عناصر الفرد "person" كما يحتاج الفرد إلى سمة رقم معرف حتى يمكن الإشارة إلية باستخدام الأسرة أو غيرها من العناصر.



الأب والأم هما الحدود التي تنطبق منها مستخدما السمات لتجنب العواقب الوخمية، لكن تظل إمكانية الواقع في مشكلة قائمة، فعلى الرغم من أنه لكل فرد أب واحد فقط وأم واحدة، وذلك من الناحية البيولوجية، فهناك العديد من الأفراد ينتمون لآباء بالتبني ومن ثم يجب ربط هؤلاء الآباء بالفرد.

وعادة ما تنقسم الأسماء إلى اسم العائلة والاسم المفترض ما يسمح بالقيام ببعض المهام مثل كتابة ورقة النمط بالأسود العريض تضم كل الأفراد الذين تنتهي أسماؤهم بالاسم هارولد.

أما الميلاد والوفاة والدفن "ومن الممكن أيضا الديانة حيث أنها أحيانا تكون الشيء الوحيد المتوفر عن فردها" فهم ينقسمون إلى تاريخ "ويمكن أن يتضمن الوقت" ومكان، وقد يكون هذا المكان CDATA أو يكون عنصر كامل. غير أنه في الواقع أحيانا لا تتوافر العناوين الكاملة للشوارع التي يرسل عليها أي مكتب بريد. ومن الأسماء الأكثر شيوعا نجد مارنت ستيرلنج كنتاكي "Mount Sterling Kentucky" أو أسماء لمزرعة إحدى الأسر المعروفة.

وعن التواريخ، فيمكن أن تخزين علي أنها CDATA، أو أن تفرق إلى يوم وشهر وسنة، وهذه الطريقة الثانية تعد أسهل من الالتزام تنسيق عام للتواريخ.

وهذا ليس بكل شئ، فلم نتناول بعد إحدى أهم المعلومات وهي الملاحظات. وقد تحتوى ملاحظة ما عن فرد معين على بيانات بسيطة أو على قصة كاملة مثل قصة "مقتل سام أتدرسون في المزرعة". وقد تكون تلك المعلومات شخصية كأن نتحدث مثلاً عن الديانة، أو طبية كان نتحدث مثلاً عن الأجداد من أودى سرطان المعدة بحياتهم. وإن كنت مولعاً بمعلومات معينة عن مثلاً أو عن التاريخ الطبي، فإنه يمكنك عمل عناصر خاصة بها، إلى جانب بعض العناصر الأخرى التي تحمل المعلومات التي عثرت عليها أثناء عملية البحث.

وهناك أشياء عدة يمكن تضمينها في عنصر PERSON كالصور على سبيل المثال، لكنني سأتوقف هنا الآن حتى يسهل فهم الفصل وننتقل الآن إلى عنصرSOURCE .

History of Kentucky المجلد الثاني ص ٣٢٥٠، ١٨٤٠، ١٨٧٥ والمصادر من هذا النوع بناء داخلي يحمل عناصر مثل BOOK, AUTHOR, VOLUME, PAGE_RANGE, YEAR إلى غير ذلك من العناصر. وتتوالى المجهودات وتضافر لمحاولة أنتاج DTD للوصف العام للكتب والمخطوطات.

أما عن العناصر في علم الأنساب تحديداً فهي أقل ترتيباً من غيرها، وإحدى أهم تلك العناصر هي الكتاب المقدس الخاص بالأسرة والذي يشتمل على تسجيلات بالميلاد والتواريخ والزيجات في مثل هذه الحالة لا تكون نسخة الكتاب المقدس أو ترجم أو ناشرة هو المهم، بل المهم هو النسخة الخاصة بالفرد الموجود بالمنزل. وهناك مثال آخر لتلك المصادر، والذي ينضح عندما تحاول ذكر ما عثرت علية في جريدة قد مر على إصدارها قرابة خميسين عام. ستكون المعلومات الموجودة بالنعي أقرب إلى الصحة، لكن لن يكون من السهل تحديد رقم الصفحة أو اسم الجريدة أو تاريخ إصدارها.

نظراً لأن تطوير تطبيق XML الخاص بالوصف العام للكتب والمخطوطات سيتعدى الفصل الواحد، ونظراً لأن تلك هي مهمة من الأفضل تركها لأمناء المكتبة المحترفين، فإنني سأكتفي بعمل عنصر SOURCE يحتوي فقط على بيانات الأحرف، وسيكون له أيضا سمة ID التي ستأخذ شكل s1, s2, s3 وهكذا، بحيث يمكن استخدام عناصر مختلفة للإشارة إلى كل عنصر على حدة وننتقل الآن إلى كتابة DTD التي تزود تطبيق XML بالوثائق.

الفرد DTD

باستخدام مراجع الكيانات الخارجية، يصبح من الممكن تخزين الأفراد في ملفات منفصلة، ثم إدراجهم بعد ذلك في العائلات أو شجرات العائلات أذن فلنبدأ DTD خاص بفرد واحد ثم ندمجه في القسم اللاحق في DTDخاص بالعائلات وشجرات العائلات.

ويكون عدة مفيداً أن يسير العمل بالعكس، أي من الأمام إلى الخلف، وذلك عند تطوير DTD، كأن تكتب أولاً تعليم XML الذي تود رؤيته مستعيناً في ذلك بمثال واحد أو مثالين، ثم تكتب بعد ذلك DTD الذي يتناسب مع البيانات. وسأتخذ جدي موئيل انجليش آندرسون مثالاً، لتوافر معلومات كافية عنة تفي بالغرض، ولأنة رحل عن دينانا منذ أمد بعيد فلا يغضب أحد لما سأقوله عنة (ستعجب كثيراً لما سوف تعرفه عند الأنساب). وهاهي المعلومات المتوافرة لدى عن صموئيل إنجليش آندرسون:

الاسم: صموئيل إنجليش آندرسون

الميلاد: ٢٥ أغسطس ١٨٧١ عرض جانبي

الوفاة: ١٠ نوفمبر ١٩١٩ ام تي ستيرلنج، كي وأي

الأب: توماس نحوروين آندرسون ١٨٤٥-١٨٨٩

الأم: ليانا (لي آنا آفي) ديمينت إنجليش (١٨٤٣-١٨٩٨)

خيارات متفرقة. ملاحظات٢١٩

كان صموئيل إنجليش آندرسون معروفاً في مونتجوميرى كاونتي بشعره الأحمر المتوهج وطبعة الحاد، فقد حدث ذات يوم أن قتل رجلاً لكن المحكمة اعتبرتها حالة دفاع عن النفس.

ولقد مات برصاص عامل في المزرعة كان قد فصله عن العمل في اليوم السابق لحادث لأنة كان يدخن في مخزن للتبغ، لكن هامب يقول أنها ربما تكون حالة دفاع عن النفس لأن صموئيل كان قد هدد بقتل العاملين الذين يدخنون في المخزن. وتناثرت الإشاعات التي تقول بأن العاملين قد حطموا رأس صموئيل، بينما تقول بيت أنه قد أتت إلى مسامعها بعض الأخبار التي تفيد بأن صموئيل قد لقي حتفه ضربا بالمناجل، غير أن هامب يرد على ذلك بأنهم لم يكونوا يستخدموا المناجل في ذلك الوقت لأنهم الوقت لا يقومون بتقطيع التبغ في شهر نوفمبر، بل يقومون فقط بتقشيره في المخازن.

والآن نعيد تنسيق ما ذكر في XML كما هو موضح في تعليمات البرمجة ٢٣-١

تعليمات البرمجة ٢٣-١: مستند XML لصموئيل إنجليش آندرسون

```
<?xml version="1.0"?>
<!DOCTYPE PERSON SYSTEM “person.dtd">
<PERSON ID="p37" SEX="M">
  <REFERENCE SOURCE="s29"/>
  <REFERENCE SOURCE="s43"/>
  <NAME>
    <GIVEN>Samuel English</GIVEN>
    <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
    <PLACE>Sideview</PLACE>
    <DATE>25 Aug 1871</DATE>
  </BIRTH>
  <DEATH>
```

```
   <PLACE>Mt. Sterling, KY</PLACE>
   <DATE>10 Nov 1919</DATE>
  </DEATH>
  <SPOUSE PERSON="p1099"/>
  <SPOUSE PERSON="p2660"/>
  <FATHER PERSON="p1035"/>
  <MOTHER PERSON="p1098"/>
  <NOTE>
   <REFERENCE SOURCE="s219"/>
   <body>
    <p>
     Samuel English Anderson was known in Montgomery County
     for his red hair and the temper that went with it. He
     did once <strong>kill a man</strong>, but the court
     found that it was in self-defense.
</p>

    <p>
     He was shot by a farm worker whom he had
     Fired the day before for smoking in a tobacco barn.
     Hamp says this may have been self-defense, because he
     Threatened to kill the workers for smoking in the barn.
     He also says old-time rumors say they mashed his head
     With a fence post. Beth heard he was cut to death with
     Machetes in the field, but Hamp says they wouldnصt be
     Cutting tobacco in November, only stripping it in the
     Barn.
    </p>
   </body>
  </NOTE>
</PERSON>
```

تم التخلص من المعلومة الواردة عن أفراد آخرين، وتم استبدالها بمراجع لـــهؤلاء الأفـــراد. وتقوم قاعدة البيانات المستخدمة لتخزين تلك المعلومة بتوفير الأرقام المعرفة، كما تصبح أرقـــام التعليق الختامي سمات SOURCE للعناصر REFERENCE وتستخدم علامـــات HTML لتعليـــم الملاحظة.

والآن لنر كيف سيكون شكل DTD لكل ذلك. أول عنصر PERSON، وقـــد يحتــوي علــى أسماء، مراجع، مواليد، وفيات، دفن، ديانات، ملاحظات، أزواج، أباء وأمهات، وسأسمح بوجـــود معلومة واحدة، وربما أقل وربما أكثر، لكل بغير ترتيب معين.

```
<!ELEMENT PERSON (NAME | REFERENCE | BIRTH | DEATH | BURIAL
  | BAPTISM | NOTE | SPOUSE | FATHER | MOTHER )*>
```

وللوهلة الأولى سيبدو غريبا عدم طلب عنصر BIRTH أو غيرها من العناصر، فلكــل يــوم ميلاد واحد، لكن لا تنسى أن ما يذكر هنا ما هو إلا معلوماتنا نحن عن هؤلاء الأفراد، فإنك كثيراً ما تكون تعرف شخصاً ما لكنك لا تعلم تحديداً بيوم أو عام ميلاده. كذلك يمكن أن تكون المصادر المتوفرة لديك متعارضة بحيث تعطي قيماً مختلفة لأيام الميلاد مثلاً أو لغيرها من المعلومات. لذا يكون من الضروري تضمين بيانات إضافية.

وعنصر PERSON يكون له سمتان: ID التي سنطلبها، وسمة SEX التي ستجعلها اختياريـــة بحيث يمكن طلبها أو إهمالها (غالباً ما تحتوي التسجيلات القديمة علـــى الأطفـــال دون تحديـــد أنواعهما، فقط أسماؤهم أحيانا لا تذكر الأسماء حتى الصور الفوتوغرافية لا توضح النوع خاصة في حالة الأطفال الذين توفوا في سنة مبكرة)

```
<!ATTLIST PERSON
  ID ID     #REQUIRED
  SEX (M | F) #IMPLIED>
```

يجب بعد ذلك إعلان عناصر الطفل. أربعة من هذه العناصر، وهي تحديداً الميـــلاد والوفـــاة والدفن والديانة، تتكون من مكان وتاريخ، ومن ثم تعد مكاناً جيداً لمرجع كيان معامل.

```
<!ENTITY % event  "(REFERENCE*, PLACE?, DATE?)*">
<!ELEMENT  BIRTH   %event;>
<!ELEMENT  BAPTISM %event;>
<!ELEMENT  DEATH   %event;>
<!ELEMENT  BURIAL  %event;>
```

ولقد أضفت أيضا عنصر REFErENCE واحد وأكثر اختياريا في البداية على الرغم مـــن أن هذا المثال ليس له SOURCE لأي معلومة سابقة. ستجد في بعض الأحيـــان مصـــادر مختلفـــة لمعلومات مختلفة خاصة بفرد واحد. وفي الواقع سأقوم بإضافة عناصر REFERENCE كنتــــائج أساسية لكل عنصر في DTD. وتكون بهذا الشكل التالي ملحق بها تعليقا في حالة عـــدم القـــدرة على تحديد ما يوجد بالمرجع بمجرد إلقاء نظرة على DTD.

```
<!- The ID number of a REFERENCE element
   that documents this entry ->
<!ELEMENT  REFERENCE  EMPTY>
<!ATTLIST  REFERENCE SOURCE NMTOKEN #REQUIRED>
```

وتحتوي سمة SOURCE هنا على رقم المصدر المتماثل، وبإضافة عناصر ISOURCEالأصلية إلى DTD الموضح بأسفل، سيصيح ذلك الرقم المعروف لعنصرSOURCE .

أما PLACE فيحتوي فقط على النص، بينما يحتوي DATE على سلسة التاريخ. ولقد اتخذت قراراً بالسماح بوجود التواريخ غير المؤكدة، الشائع استخدامها في علم الأنساب، كأن نقول مثلاً في حوالي عام ١٨٧٦ أو ما قبل عام ١٩٢٠ وذلك في مقابل طلب سنة وتاريخ وشهر منفصلتين.

```
<!ELEMENT  PLACE (#PCDATA)>
<!ELEMENT  DATE  (#PCDATA)>
```

تتصل السمات FATHR وSPOUSE MOTHER الخاص بعنصر PERSON وذلك عبر السمة PERSON مرة أخرى نقول أن هذه تعتبر فرصة جيدة لاستخدام مرجع كيان معامل.

```
<!ENTITY % personref "PERSON NMTOKEN #REQUIRED">
<!ELEMENT  SPOUSE  EMPTY>
<!ATTLIST  SPOUSE  %personref;>
<!ELEMENT  FATHER  EMPTY>
<!ATTLIST  FATHER  %personref;>
<!ELEMENT  MOTHER  EMPTY>
<!ATTLIST  MOTHER  %personref;>
```

وإنه لمثالي أن تكون سمة PERSON من نوع IDREF، ولكن طالما أن الفرد المحدد يمكن تواجده في ملف آخر، فإن أفضل ما يمكن عملة هو طلب النوع NMTOKEN.

قد يحتوي عنصر NAME على أي عدد من عناصر REFERENCE لكنة يحتوي على عدد من العناصر SURNAME وGIVEN، قد يكون هذا العدد واحد أو صفر، أي لا يوجد ويحتوي كل من هذين العنصرين على نص.

```
<!ELEMENT  NAME    (REFERENCE*, GIVEN?, SURNAME?)>
<!ELEMENT  GIVEN   (#PCDATA)>
<!ELEMENT  SURNAME (#PCDATA)>
```

The NOTE element may contain an arbitrary amount of text. Some standard markup would be useful here. The easiest solution is to adopt the XHTML DTD introduced in Chapter 20, Reading Document Type Definitions. It's not necessary to rewrite it all. Simply use a parameter reference entity to import it. We'll allow each NOTE to contain zero or more REFERENCE elements and a single body element.

```
<!ENTITY % xhtml SYSTEM "xhtml/XHTML1-s.dtd">
%xhtml;
<!ELEMENT  NOTE   (REFERENCE*, body)>
```

وتوضح هذه السطور الثلاث تعين تعليم HTML 4.0 بأكمله فلا يكون هناك داع لإصدار تعينا خاص بك بل يمكنك الاعتماد علي تعين العلامات HTML المعروف والمدعم. وأنا لم أذكـر رأس الصفحة هنا، لكنة من السهل تضمينها وذلك بوضع html بدلاً من body المذكورة بأعلى. "وأنـا لم اذكر رأس الصفحة لأن ذلك يتطلب تضمين العناصر head وtitle، الأمـــر الــذي لا يعـد ضروريات في هذه الحالة" ويفترض ذلك أن يكون الملف XHTML1-s.dtd موجوداً في الملــف النسبي URL xhtml/XHTML1-s.dtd على الرغم من انه يسهل وضعه في أي مكان آخر. كما يمكن أيضا استخدام URL مطلق على موقع ويــب W3C http://www.w3.org/TR/xhtml-modularization/DTD/XHTML، على الرغم من أنني لا أفضل أن تعتمد الملفات على موقــع ويب لا أستطيع التحكم في توفيره. وتوضح تعليمات البرمجة ٢٣-٢ DTD الكامل.

تعليمات البرمجة ٢٣-٢: person.dtd كاملة: person.dtd

```
<!ELEMENT PERSON ( NAME | REFERENCE | BIRTH | DEATH | BURIAL
          | BAPTISM | NOTE | FATHER | MOTHER | SPOUSE )* >
<!ATTLIST PERSON ID  ID #REQUIRED>

<!-M means male, F means female ->
<!ATTLIST PERSON SEX (M | F) #IMPLIED>

<!- The ID number of a SOURCE element that documents
   this entry ->
<!ELEMENT  REFERENCE  EMPTY>
<!ENTITY % sourceref "SOURCE NMTOKEN #REQUIRED">
<!ATTLIST  REFERENCE %sourceref;>

<!ENTITY % event   "(REFERENCE*, PLACE?, DATE?)">
<!ELEMENT  BIRTH   %event;>
<!ELEMENT  BAPTISM %event;>
<!ELEMENT  DEATH   %event;>
<!ELEMENT  BURIAL  %event;>

<!ELEMENT  PLACE   (#PCDATA)>
<!ELEMENT  DATE    (#PCDATA)>

<!ENTITY % personref :"PERSON NMTOKEN #REQUIRED">
```

```
<!ELEMENT  SPOUSE  EMPTY>
<!ATTLIST  SPOUSE  %personref;>
<!ELEMENT  FATHER  EMPTY>
<!ATTLIST  FATHER  %personref;>
<!ELEMENT  MOTHER  EMPTY>
<!ATTLIST  MOTHER  %personref;>

<!ELEMENT  NAME    (GIVEN?, SURNAME?)>
<!ELEMENT  GIVEN   (#PCDATA)>
<!ELEMENT  SURNAME (#PCDATA)>

<!ENTITY % xhtml SYSTEM "xhtml/XHTML1-s.dtd">
%xhtml;
<!ELEMENT  NOTE   (REFERENCE*, body)>
```

DTD الأسرة

الخطوة التالية هي كتابة DTD للعائلة ولنبدأ بعينة مستند XML كما هو موضح في تعليمات البرمجة٢٣-٣.

تعليمات البرمجة ٢٣-٣: مستند XML لعائلة صموئيل إنجليش آندرسون

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE FAMILY SYSTEM "family.dtd">
<FAMILY>
  <HUSBAND PERSON="p37"/>
  <WIFE    PERSON="p1099"/>
  <CHILD   PERSON="p23"/>
  <CHILD   PERSON="p36"/>
  <CHILD   PERSON="p1033"/>
  <CHILD   PERSON="p1034"/>
  <MARRIAGE>
    <PLACE>Cincinatti, OH</PLACE>
    <DATE>15 Jul 1892</DATE>
  </MARRIAGE>
</FAMILY>
```

كل ما تحتاجه هنا مراجع لأفراد، العائلة وليس أفراد العائلة أنفسهم مرة أخرى توفر قاعدة البيانات المخزن بها المعلومة الأرقام المعرفة للمرجع PERSON، والتي لا تعد قيمها الحقيقية ذات قيمة طالما أنها فريدة وثابتة.

الآن بعد إتاحة عائلة نموذجية، يجب إعداد DTD لكل العائلات كما هو موضح في تعليمات البرمجة ٢٣-٤ لكن تذكر إضافة بعض العناصر التي قد تحتاجها بعض العائلات وليست مذكورة في هذا المثال، كالطلاق مثلا. وسيقوم مرجع كيان معامل بسحب الإعلانات من DTD الفرد لتعليمات البرمجة ٢٣-٢

تعليمات البرمجة ٢٣-٤: DTD لوصف العائلة family.dtd

```
<!ENTITY % person SYSTEM "person.dtd">
%person;

<!ELEMENT FAMILY (REFERENCE*, HUSBAND?, WIFE?, CHILD*,
          MARRIAGE*, DIVORCE*, NOTE*)>

<!ELEMENT  HUSBAND  EMPTY>
<!ATTLIST  HUSBAND  %personref;>
<!ELEMENT  WIFE     EMPTY>
<!ATTLIST  WIFE     %personref;>
<!ELEMENT  CHILD    EMPTY>
<!ATTLIST  CHILD    %personref;>
<!ELEMENT  DIVORCE  %event;>
<!ELEMENT  MARRIAGE %event;>
```

وأنا لا افترض وجود أكثر من HUSBAND واحد أو WIFE واحدة لكل عنصر FAMILY، وهو الافتراض القياسي في علم الأنساب، حتى في البلاد التي تسمح بتعدد الزيجات، فسمح بذلك باستيعات كل الأطفال. لكنك ستلحظ أنه عند تزويد علم الأنساب في المجتمعات التي تسمح بتعدد الزوجات بالمستندات، ستجد أنة قد سيظهر HUSBAND نفسه أكثر من مرة في عناصر FAMILY المتعددة، وكذلك الحال بالنسبة للمجتهدات التي تسمح بتعدد أزواج، حيث ستجد أنة قد تظهر WIFE واحدة أكثر من مرة في عناصر FAMILY المتعددة. وإذا نحينا تداخل التواريخ معاً، سنجد أن هذا هو نفس الإجراء الذي يتخذ عند تزويد سلسلة من الزيجات بالمستندات وبالطبع، لا يوجد شئ لإنجاب أطفال "ولا حتى في علم لأحياء".

وعامة، يمكن وصف هذا البرامج بالمرونة، حيث يسمح لعنصر FAMILY إن يحتوي على عناصر PERSON بذاتها بدلاً من أن يحتوي على مجرد إشارة لها، وذلك يتطلب بدوره نسخ البيانات ونقلها إلى عناصر وملفات مختلفة. غير أن الشيء الوحيد الذي لا يستطيع DTD القيام به على خير وجه هو الزيجات بين أفراد من نفس النوع، لكن يمكن تناول بتغير إعلان FAMILY ليصبح كالآتي.

```
<!ELEMENT FAMILY (((HUSBAND, WIFE) | (HUSBAND, HUSBAND?)
                 | (WIFE, WIFE?)), MARRIAGE*, DIVORCE*, CHILD*)>
```

قد يبدو السماح بوجود زيجات أو طلقات متعددة غريباً بعض الشيء لكنة موجود بالفعل, فوالدة زوجتي على سبيل المثال تزوجت من والد زوجتي وانفصلت عنه ثلاث مرات منفصلة. لذا فعلى الرغم من الزواج أكثر من مرة من نفس الشخص لا يعد أمراً شائعاً لكنه موجود بالفعل.

المصدر DT

أما SOURCE فهو العنصر الثالث والأخير، الذي يستخدم مع البناء الداخلي البسيط، غير أن تخزين DTD في ملف منفصل يسهل إضافة بناء آخر لها فيما بعد. وتبدو بعض عناصر SOURCE النموذجية كالآتي.

```
<SOURCE ID="s218">Hamp Hoskins interview, 11-28-1996</SOURCE>
<SOURCE ID="s29">English-Demint Anderson Bible</SOURCE>
<SOURCE ID="s43">Anderson Bible</SOURCE>
<SOURCE ID="s43">
  Letter from R. Foster Adams to Beth Anderson, 1972
</SOURCE>
<SOURCE ID="s66">
  Collinصs History of Kentucky, Volume II, p.325, 1840, 1875
</SOURCE>
```

وقد يكون للعنصر SOURCE أكثر من بناء داخلي، بحيث يسير العمل في أكثر مكان لا نتاج DTD شامل لكل المعلومات البيلوغرافية مع عناصر خاصة بالمقالات والمؤلفين، والصفحات وتواريخ النشر وغيرها. لكن بوجه عام يعد هذا الموضوع معقداً بعض الشيء، بالإضافة إلى أن استخدامه في علم الأنساب يكون مختلفاً عن استخدامه في معظم المجالات الأخرى. وتعتبر النسخة الخاصة للكتاب المقدس للعائلة أو لجريدة بها حاشية مكتوبة بخط اليد أكثر أهمية من المؤلف القياسي والعنوان وناشر البيانات التي تتميز جميعها بأنها أعم وأشمل والتي تستخدم في وصف الكتب والمخطوطات.

ونظراً لأن تطوير تطبيق XML الخاص بوصف الكتب والمخطوطات قد يحتاج لأكـثر مـن فصل للحديث عنة، ونظراً لأنها مهمة يفصل إسنادها إلى أمناء المكتبـات المحـترفين، فـإنني سأكتفي بأن اجعل عنصر SOURCE يحتوي فقط على بيانات الأحرف وسيكون له سمه ID على شكل s1 وs2 وs3 الخ، حتى يمكن الإشارة إلى كل مصدر على حده باستخدام عناصر مختلفـة. وتوضح تعليمات البرمجة ٢٣-٥ أبسط أشكال DTD للمصادر.

تعليمات البرمجة ٢٣-٥: مصدر DTD بسيط: SOURCE DTD

```
<!ELEMENT  SOURCE  (#PCDATA)>
<!ATTLIST  SOURCE ID  ID #REQUIRED>
```

شجرة العائلة DTD

أصبح الآن ممكنا دمج العائلات المختلفة والأفراد والمصادر في تجميع واحد يشمل كـل فـرد. وسأطلق على العنصر المصدر لـهذا المسـتند اسـم FAMILY_TREE، وسيشـمل عنـاصر PERSON، FAMILY وSOURCE لكن دون ترتيب معين.

```
<!ELEMENT FAMILY_TREE (PERSON | FAMILY | SOURCE)*>
```

أصبح الآن من الضروري أعاده إعلان عناصر PERSON وSOURCE FAMILY ونتائجهم، ويمكن بدلاً من ذلك استيرادها وذلك باستيراد العائلة وDTD المصادر مع مراجع كيانات معاملة ثم تستورد العائلة وDTD العائلة بدورها DTD الفرد.

```
<!ENTITY % family SYSTEM "family.dtd">
%family;
<!ENTITY % source SYSTEM "source.dtd">
%source;
```

وعند بلوغ هذه المرحلة، استخدم الأنواع ID الأصلية للأزواج والآباء والمراجع بـدلاً مـن الأنواع NMTOKEN. وذلك لأن عنصر FAMILY الذي يكون جزءً من FAMILY_TREE يجب أن يشتمل على كل عناصر PERSON اللازمة. ويمكنك القيام بذلك بإلغاء إعلانـات الكيانـات المعاملة personref وsourceref في DTD لشجرة العائلة.

```
<!ENTITY % personref "PERSON IDREF #REQUIRED">
<!ENTITY % sourceref "SOURCE IDREF #REQUIRED">
```

هذا كل ما تحتاج إليه. وأي شيء أخر ستجده في الفـرد المسـتورد وفـي DTD للعـائلات وتوضح تعليمات البرمجة ٢٣-٦ شجرة العائلة DTD، سينما توضح تعليمـات البرمجـة ٢٣-٧ مستندا لشجرة العائلة كاملة يصم أحد عشر فرداً وثلاث أسر وسبعة مصادر.

تعليمات البرمجة ٢٣-٦: شجرة العائلة DTD :family tree DTD

```
<!ENTITY % personref "PERSON IDREF #REQUIRED">
<!ENTITY % sourceref "SOURCE IDREF #REQUIRED">

<!ENTITY % family SYSTEM "family.dtd">
%family;

<!ENTITY % source SYSTEM "source.dtd">
%source;

<!ELEMENT FAMILY_TREE (SOURCE | PERSON | FAMILY )*>
```

تعليمات البرمجة ٢٣-٧: مستند XML لشجرة كاملة

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE FAMILY_TREE SYSTEM "familytree.dtd">
<FAMILY_TREE>

 <PERSON ID="p23" SEX="M">
  <REFERENCE SOURCE="s44"/>
  <FATHER PERSON="p37"/>
  <MOTHER PERSON="p1099"/>
  <NAME>
   <GIVEN>Judson McDaniel</GIVEN>
   <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
   <PLACE>Montgomery County, KY, 1893</PLACE>
   <DATE>19 Jul 1894</DATE>
  </BIRTH>
  <DEATH>
   <PLACE>Mt. Sterling, KY</PLACE>
   <DATE>27 Apr 1941</DATE>
  </DEATH>
  <NOTE><body>
   <p>Agriculture College in Iowa</p>
```

```
    <p>Farmer</p>
    <p>32nd degree Mason</p>
    <p>
     He shot himself in the pond in the back of Sideview
     when he found that he was terminally ill. It has also
     been claimed that he was having money and wife
     troubles. (He and Zelda did not get along and he was
     embarrassed to have  married her.) It has further been
     claimed that this was part of the Anderson family
     curse.
    </p>
   </body></NOTE>
  </PERSON>

  <PERSON ID="p36" SEX="F">
   <REFERENCE SOURCE="s43"/>
   <FATHER PERSON="p37"/>
   <MOTHER PERSON="p1099"/>
   <NAME>
    <GIVEN>Mary English</GIVEN>
    <SURNAME>Anderson</SURNAME>
   </NAME>
   <BIRTH>
    <PLACE>August 4, 1902?, Sideview, KY</PLACE>
    <DATE>8 Apr 1902</DATE>
   </BIRTH>
   <DEATH>
<PLACE>Mt. Sterling, KY</PLACE>
    <DATE>19 Dec 1972</DATE>
   </DEATH>
  </PERSON>

  <PERSON ID="p37" SEX="M">
   <REFERENCE SOURCE="s29"/>
   <REFERENCE SOURCE="s43"/>
   <FATHER PERSON="p1035"/>
   <MOTHER PERSON="p1098"/>
   <NAME>
```

```
    <GIVEN>Samuel English</GIVEN>
    <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
    <PLACE>Sideview</PLACE>
    <DATE>25 Aug 1871</DATE>
  </BIRTH>
  <DEATH>
    <PLACE>Mt. Sterling, KY</PLACE>
    <DATE>10 Nov 1919</DATE>
  </DEATH>
  <NOTE>
    <body>
      <p>
        Samuel English Anderson was known in Montgomery
        County for his red hair and the temper that went
        with it. He did once <strong>kill a man</strong>,
        but the court found that it was in self-defense.
      </p>

      <p>
        He was shot by a farm worker whom he had
        fired the day before for smoking in a tobacco barn.
        Hamp says this may have been self-defense, because he
        threatened to kill the workers for smoking in the
        barn. He also says old-time rumors say they mashed
        his head with  a fence post. Beth heard he was cut to
        death with machetes in the field, but Hamp says they
        wouldnصt be cutting tobacco in November, only
        stripping it in the barn.
      </p>
    </body>
  </NOTE>

</PERSON>

<PERSON ID="p1033" SEX="M">
  <REFERENCE SOURCE="s43"/>
```

```
  <FATHER PERSON="p37"/>
  <MOTHER PERSON="p1099"/>
  <NAME>
   <GIVEN>Thomas Corwin</GIVEN>
   <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
   <DATE>16 Jan 1898</DATE>
  </BIRTH>
  <DEATH>
   <PLACE>Probably Australia</PLACE>
  </DEATH>
  <NOTE>
   <body><p>
     Corwin fought with his father and then left home.
     His last letter was from Australia.
    </p></body>
  </NOTE>
</PERSON>

<PERSON ID="p1034" SEX="M">
  <REFERENCE SOURCE="s43"/>
  <FATHER PERSON="p37"/>
  <MOTHER PERSON="p1099"/>
  <NAME>
   <GIVEN>Rodger French</GIVEN>
   <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
   <DATE>26 Nov 1899</DATE>
  </BIRTH>
  <DEATH>
   <PLACE>Birmingham, AL</PLACE>
  </DEATH>
  <NOTE>
   <body><p>
    Killed when the car he was driving hit a pig in the
    road; Despite the many suicides in the family, this is
```

```
    the only known sowicide.
   </p></body>
  </NOTE>
 </PERSON>

 <PERSON ID="p1035" SEX="M">
  <NAME>
   <GIVEN>Thomas Corwin</GIVEN>
   <SURNAME>Anderson</SURNAME>
  </NAME>
  <BIRTH>
   <DATE>24 Aug 1845</DATE>
  </BIRTH>
  <DEATH>
   <PLACE>Mt. Sterling, KY</PLACE>
   <DATE>18 Sep 1889</DATE>
</DEATH>
  <NOTE>
   <body>
    <p>Yale 1869 (did not graduate)</p>
    <p>Breeder of short horn cattle</p>
    <p>He was named after an Ohio senator. The name Corwin
     is from the Latin <i>corvinus</i> which means
    <i>raven</i> and is akin to <i>corbin</i>/<i>corbet</i>.
     In old French it was <i>cord</i> and in Middle English
     <i>Corse</i> which meant <i>raven</i> or <i>cow</i>.
    </p>
    <p>Attended Annapolis for one year, possibly to
     avoid service in the Civil War.</p>
    <p>He farmed the old Mitchell farm
     and became known as a leading short horn breeder.
     He suffered from asthma and wanted to move to
     Colorado in 1876 to avoid the Kentucky weather, but
     he didn't.
      </p>
   </body>
  </NOTE>
 </PERSON>
```

```
<PERSON ID="p1098" SEX="F">
 <REFERENCE SOURCE="s29"/>
 <NAME>
  <GIVEN>LeAnah (Lee Anna, Annie) DeMint</GIVEN>
  <SURNAME>English</SURNAME>
 </NAME>
 <BIRTH>
  <PLACE>Louisville, KY</PLACE>
  <DATE>1 Mar 1843</DATE>
 </BIRTH>
 <DEATH>
  <REFERENCE SOURCE="s16"/>
  <PLACE>acute Bright's disease, 504 E. Broadway</PLACE>
  <DATE>31 Oct 1898</DATE>
 </DEATH>
 <NOTE>
  <body>
   <p>Writer (pseudonymously) for Louisville Herald</p>
   <p>Ann or Annie was from Louisville. She wrote under
     an assumed name for the Louisville Herald.</p>
  </body>
 </NOTE>
</PERSON>

<PERSON ID="p1099" SEX="F">
 <REFERENCE SOURCE="s39"/>
 <FATHER PERSON="p1100"/>
 <MOTHER PERSON="p1101"/>
 <NAME>
  <GIVEN>Cora Rucker (Blevins?)</GIVEN>
  <SURNAME>McDaniel</SURNAME>
 </NAME>
 <BIRTH>
  <DATE>1 Aug 1873</DATE>
 </BIRTH>
 <DEATH>
  <REFERENCE SOURCE="s41"/>
```

```
   <REFERENCE SOURCE="s60"/>
   <PLACE>Sideview, bronchial trouble TB</PLACE>
   <DATE>21 Jul 1909</DATE>
  </DEATH>
  <NOTE>
   <body>
    <p>She was engaged to General Hood of the Confederacy,
    but she was seeing Mr. Anderson on the side. A servant
    was posted to keep Mr. Anderson away. However the girl
    fell asleep, and Cora eloped with Mr. Anderson.</p>
   </body>
  </NOTE>
</PERSON>

<PERSON ID="p1100" SEX="M">
  <NAME>
   <GIVEN>Judson</GIVEN>
   <SURNAME>McDaniel</SURNAME>
  </NAME>
  <BIRTH>
   <DATE>21 Feb 1834</DATE>
  </BIRTH>
  <DEATH>
   <DATE>9 Dec 1905</DATE>
  </DEATH>
</PERSON>

<PERSON ID="p1101" SEX="F">
  <NAME>
   <GIVEN>Mary E.</GIVEN>
   <SURNAME>Blevins</SURNAME>
  </NAME>
  <BIRTH>
   <DATE>1847</DATE>
  </BIRTH>
  <DEATH>
   <DATE>1886</DATE>
  </DEATH>
```

```
   <BURIAL>
     <PLACE>Machpelah Cemetery, Mt. Sterling KY</PLACE>
   </BURIAL>
 </PERSON>

<PERSON ID="p1102" SEX="M">
   <REFERENCE SOURCE="s29"/>
   <NAME>
     <GIVEN>John Jay (Robin Adair )</GIVEN>
     <SURNAME>Anderson</SURNAME>
   </NAME>
   <BIRTH>
     <REFERENCE SOURCE="s43"/>
     <PLACE>Sideview</PLACE>
     <DATE>13 May 1873</DATE>
   </BIRTH>
   <DEATH>
     <DATE>18 Sep 1889  </DATE>
   </DEATH>
   <NOTE><body><p>
     Died of flux. Rumored to have been killed by his brother.
   </p></body></NOTE>
 </PERSON>

 <FAMILY ID="f25">
   <HUSBAND PERSON="p37"/>
   <WIFE PERSON="p1099"/>
   <CHILD PERSON="p23"/>
   <CHILD PERSON="p36"/>
   <CHILD PERSON="p1033"/>
   <CHILD PERSON="p1034"/>
 </FAMILY>

 <FAMILY ID="f732">
   <HUSBAND PERSON="p1035"/>
   <WIFE PERSON="p1098"/>
   <CHILD PERSON="p1102"/>
   <CHILD PERSON="p37"/>
```

```
</FAMILY>

<FAMILY ID="f779">
 <HUSBAND PERSON="p1102"/>
</FAMILY>

<SOURCE ID="s16">newspaper death notice in purse</SOURCE>
<SOURCE ID="s29">English-Demint Anderson Bible</SOURCE>
<SOURCE ID="s39">
 Judson McDaniel &amp; Mary E. Blevins Bible
</SOURCE>
<SOURCE ID="s41">
 Cora McDaniel obituary, clipping from unknown newspaper
</SOURCE>
<SOURCE ID="s43">Anderson Bible</SOURCE>
<SOURCE ID="s44">
 A Sesquicentenial History of Kentucky
 Frederik A. Wallis &amp; Hambleon Tapp, 1945,
 The Historical Record Association, Hopkinsville, KY
</SOURCE>
<SOURCE ID="s60">
 Interview with Ann Sandusky, May 1996
</SOURCE>

</FAMILY_TREE>
```

سيم ورقة نمط لشجرة العائلة

ظم مستند شجرة العائلة كملف بيانات لا كبيانات سردية ولتحصل على جيد للمستند عليك أولاً ادة تنظيم المحتويات قبل عرضها، وبما أن CSS لا يقدر على القيام تبلك المهمة، يتم الاستعانة رقة نمط XSL.

ومن الفضل أن تبدأ تلك المهمة بنقطة اللقاء الجذرية "التوصيل الأصلية"، والتي تستبدل هنا عناصر html و head و body. ولاستكمال المعالجة يتم تطبق القوالب على العنصر الجذري FAMILY_TRE.

```
<xsl:template match="/">
 <html>
  <head>
```

```
    <title>Family Tree</title>
  </head>
  <body>
    <xsl:apply-templates select="FAMILY_TREE"/>
  </body>
 </html>
</xsl:template>
```

وتقسم قاعدة القالب لعنصر FAMILY_TREE المستند إلى ثـــلاث أجـــزاء، الأول للعـــائلات والثاني للأفراد والثالث للمصادر، ويتم تطبيق القوالب على كل جزء على حدة.

```
<xsl:template match="FAMILY_TREE">

   <h1>Family Tree</h1>

   <h2>Families</h2>
   <xsl:apply-templates select="FAMILY"/>

   <h2>People</h2>
   <xsl:apply-templates select="PERSON"/>

   <h2>Sources</h2>
   <ul>
    <xsl:apply-templates select="SOURCE"/>
   </ul>

  </xsl:template>
```

أما عن قاعدة SOURCE فهي بسيطة للغاية، فيغلف كل مصدر عنصر li، وترفق ID الخاص بالمصـادر باستخدام السمة name لعنصر a HTML مما يسمح بتوجيه إسنادا ترافقيـــه مباشـــرة للمصدر كالآتي:

```
<xsl:template match="SOURCE">

  <li>
   <xsl:element name="a">
    <xsl:attribute name="name">
     <xsl:value-of select="@ID"/>
    </xsl:attribute>
    <xsl:value-of select="."/>
   </xsl:element>
```

```
    </li>

  </xsl:template>
```

وبالنسبة لعنصر PERSON فهو أكثر تعقيداً، لذا سنقوم بتفكيكه إلى قواعد قوالب عـــدة. ثـــم ستنتقى قاعدة القالب PERSON الأجزاء المستقلة، وتنسيق الأجزاء الأخرى غـــير المعقـــدة، ثـــم ستطبق القوالب الأخرى على باقي الجزاء. وسيوضع الاسم في رأس الصفحة h3 ويكون محاطـــاً بارتساء HTML ويكون name الخاص به هو ID الفرد. أما عن عناصر BIRTH, DEATH, BAPTISM, and BURIAL فهي يتم تنسيقها كعناصر قائمة كالآتي:

```
  <xsl:template match="PERSON">

    <h3>
      <xsl:element name="a">
        <xsl:attribute name="name">
          <xsl:value-of select="@ID"/>
        </xsl:attribute>
      <xsl:value-of select="NAME"/>
      </xsl:element>
    </h3>

    <ul>
      <xsl:if test="BIRTH">
        <li>Born: <xsl:value-of select="BIRTH"/></li>
      </xsl:if>
      <xsl:if test="DEATH">
        <li>Died: <xsl:value-of select="DEATH"/></li>
      </xsl:if>
      <xsl:if test="BAPTISM">
        <li>Baptism: <xsl:value-of select="BAPTISM"/></li>
      </xsl:if>
      <xsl:if test="BURIAL">
        <li>Burial: <xsl:value-of select="BURIAL"/></li>
      </xsl:if>
      <xsl:apply-templates select="FATHER"/>
      <xsl:apply-templates select="MOTHER"/>
    </ul>

    <p>
```

```
    <xsl:apply-templates select="NOTE"/>
   </p>

  </xsl:template>
```

كما تتغير أيضاً عناصر FATHER وMOTHER عناصر قائمة، لكن يجب ربطها بـــالأفراد الذين تشير إليهم تلك العناصر، ويقوم بهذه المهمة اثنان من قواعد القالب، هما:

```
 <xsl:template match="FATHER">
   <li>
    <xsl:element name="a">
     <xsl:attribute name="href">
      #<xsl:value-of select="@PERSON"/>
     </xsl:attribute>
     Father
    </xsl:element>
   </li>
  </xsl:template>

  <xsl:template match="MOTHER">
   <li>
    <xsl:element name="a">
     <xsl:attribute name="href">
      #<xsl:value-of select="@PERSON"/>
     </xsl:attribute>
     Mother
    </xsl:element>
   </li>
  </xsl:template>
```

أخر ما يجب عليك عملة هو تنسيق عناصر PERSON لنسخ محتويات NOTE في المســـتند النهائي. وبما أن body الخاص NOTE يستخدم علامات HTML قياسية لا تحتاج للتغير، فإنـــه في هذه الحالة يكون عنصر xsl:copy مهماً وأولى هذه القواعد تقوم بنسخ العنصـــر الأساســـي ومحتوياته:

```
 <xsl:template match="body | body//*">
  <xsl:copy>
   <xsl:apply-templates select="*|@*|comment()|pi()|text()"/>
  </xsl:copy>
 </xsl:template>
```

ثم تقوم قاعدة القالب لعناصر FAMILY بتسجيل اسم كل فرد في العائلة كعنصر قائمـــة فـــي قائمة غير مرتبة. وسيتم ربط كل فرد بالوصف الخاص وذلك عن طريق القواعد التالية:

```
<xsl:template match="FAMILY">
  <ul>
    <xsl:apply-templates select="HUSBAND"/>
    <xsl:apply-templates select="WIFE"/>
    <xsl:apply-templates select="CHILD"/>
  </ul>
</xsl:template>

<xsl:template match="HUSBAND">
  <li>Husband: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>

<xsl:template match="WIFE">
  <li>Wife: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>

<xsl:template match="CHILD">
  <li>Child: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>
```

والحيلة التي تستخدمها تلك القواعد هي إدراج بيانات من عنصر واحد (PERSON) في قالب لعناصر أخرى مختلفة (HUSBAND, WIFE, CHILD)، ثم يستخدم الرقم المعـــرف PERSON المخزن في سمة HUSBAND/WIFE/CHILD PERSON لتعين موضـــع عنصـــر PERSON الصحيح، ثم يتم اختيار نتيجتها NAME.

تعليمات البرمجة ٢٣-٨ هي ورقة النمط النهائية لشجرة العائلـــة ويوضـــح الشـــكل ٢٣-٢ المستند بعد تحويلة إلى HTML وتحميلة في Netscape Navigator.

تعليمات البرمجة ٢٣–٨: ورقة نمط لشجرة العائلة الكاملة

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">

 <xsl:template match="/">
  <html>
   <head>
    <title>Family Tree</title>
   </head>
   <body>
    <xsl:apply-templates select="FAMILY_TREE"/>
   </body>
  </html>
 </xsl:template>

 <xsl:template match="FAMILY_TREE">

  <h1>Family Tree</h1>

  <h2>Families</h2>
  <xsl:apply-templates select="FAMILY"/>

  <h2>People</h2>
  <xsl:apply-templates select="PERSON"/>

  <h2>Sources</h2>
  <ul>
   <xsl:apply-templates select="SOURCE"/>
  </ul>

 </xsl:template>
```

```
<xsl:template match="PERSON">

  <h3>
    <xsl:element name="a">
      <xsl:attribute name="name">
        <xsl:value-of select="@ID"/>
      </xsl:attribute>
    <xsl:value-of select="NAME"/>
    </xsl:element>
  </h3>

  <ul>
    <xsl:if test="BIRTH">
      <li>Born: <xsl:value-of select="BIRTH"/></li>
    </xsl:if>
    <xsl:if test="DEATH">
      <li>Died: <xsl:value-of select="DEATH"/></li>
    </xsl:if>
    <xsl:if test="BAPTISM">
      <li>Baptism: <xsl:value-of select="BAPTISM"/></li>
    </xsl:if>
    <xsl:if test="BURIAL">
      <li>Burial: <xsl:value-of select="BURIAL"/></li>
    </xsl:if>
    <xsl:apply-templates select="FATHER"/>
    <xsl:apply-templates select="MOTHER"/>
  </ul>

  <p>
    <xsl:apply-templates select="NOTE"/>
</p>

 </xsl:template>
```

```
<xsl:template match="FATHER">
 <li>
  <xsl:element name="a">
   <xsl:attribute name="href">
    #<xsl:value-of select="@PERSON"/>
   </xsl:attribute>
   Father
  </xsl:element>
 </li>
</xsl:template>

<xsl:template match="MOTHER">
 <li>
  <xsl:element name="a">
   <xsl:attribute name="href">
    #<xsl:value-of select="@PERSON"/>
   </xsl:attribute>
   Mother
  </xsl:element>
 </li>
</xsl:template>

<xsl:template match="body | body//*">
 <xsl:copy>
  <xsl:apply-templates select="*|@*|comment()|pi()|text()"/>
 </xsl:copy>
</xsl:template>

<xsl:template match="SOURCE">

 <li>
  <xsl:element name="a">
   <xsl:attribute name="name">
    <xsl:value-of select="@ID"/>
```

```
      </xsl:attribute>
      <xsl:value-of select="."/>
    </xsl:element>
  </li>

</xsl:template>

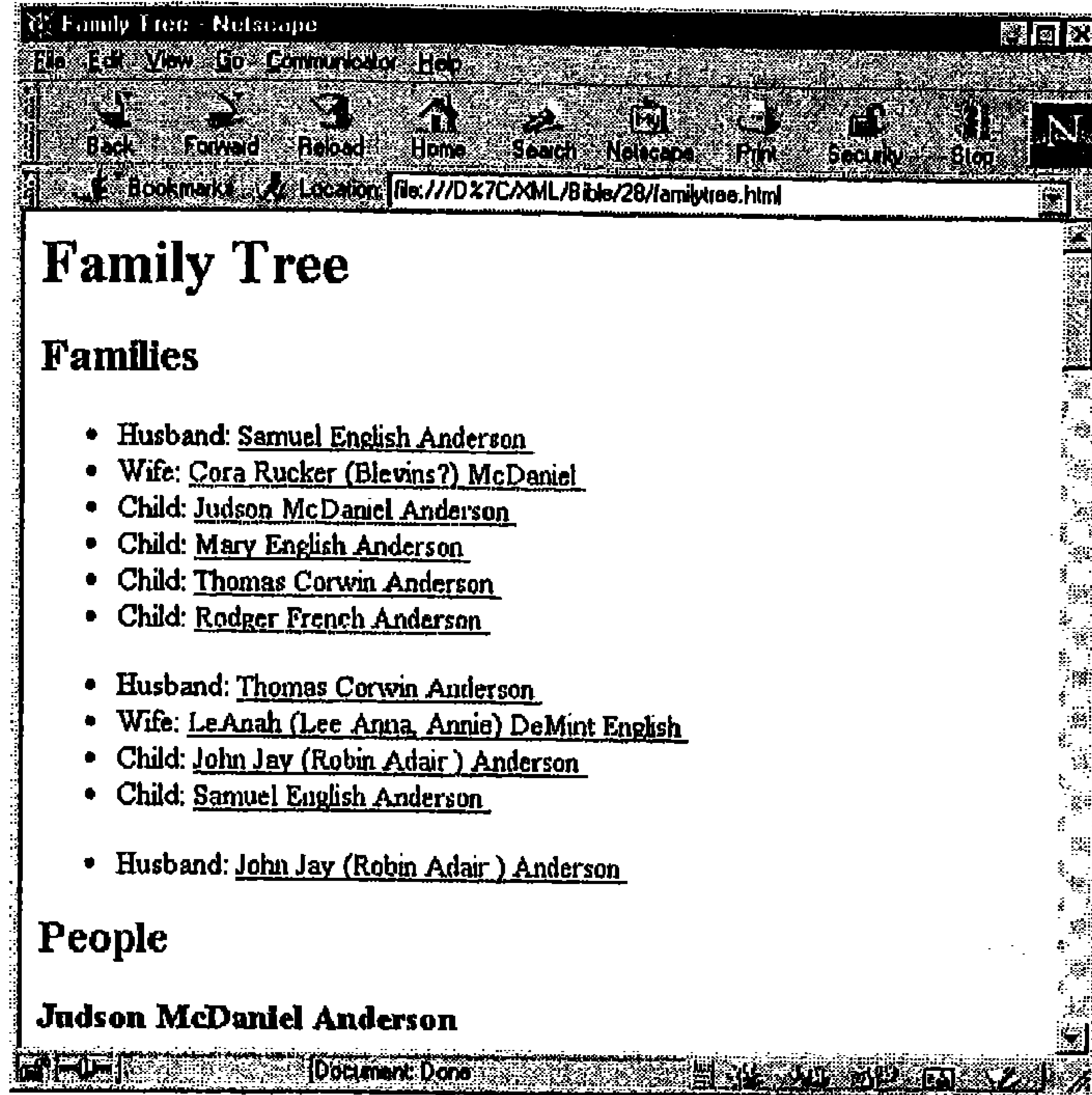
<xsl:template match="FAMILY">
  <ul>
    <xsl:apply-templates select="HUSBAND"/>
    <xsl:apply-templates select="WIFE"/>
    <xsl:apply-templates select="CHILD"/>
  </ul>
</xsl:template>

<xsl:template match="HUSBAND">
  <li>Husband: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>

<xsl:template match="WIFE">
  <li>Wife: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>

<xsl:template match="CHILD">
  <li>Child: <a href="#{@PERSON}">
    <xsl:value-of select="id(@PERSON)/NAME"/>
  </a></li>
</xsl:template>

</xsl:stylesheet>
```



الشكل ٢٣-٢ شجرة العائلة بعد تحويلها إلى HTML

خلاصة

رأيت في هذا الفصل تطوير تطبيق XML خاص بعلم الأنساب عرفت الآتي:

- ابدأ دائما تطبيق XML الجديد بتحديد المجال الذي تصفه.
- حاول تحديد العناصر الرئيسية للمجال، وأي شيء أخر إما أن يكون في محتوى تلك العناصر أو يكون سمة لها.
- حاول تجنب تضمين نفس البيانات في أكثر من مكان. استخدام السمات ID وIDREF لا.
- قامة إشارات من عنصر إلى أخر.
- ولا تنس الحالات الخاصة، فلا تستند في تصميمك كله على الحالات الواضح.
- استخدام كيانات المعلمة لدمج DTD واحدة كاملة.

يمكن أن تستنتج من ذلك الهيكل الرئيسي XML Bible. فابدأ الآن واكتب مستندات XML الخاصة بك! وستوفر الأجزاء العدة التالية معلومات مرجعية مفيدة ومتنوعة، إلى جانب المواصفة الرسمية XML 1.0.

الملاحق

مادة مرجع XML

يحتوي هذا الملحق على المادة التي يتكون منها مرجع XML وهي تنقسم إلى ثلاث أجزاء رئيسية:

١- XML BNF Grammar.

٢- Well-Formedness Constraints.

٣- Validity Constraints.

يوضح مقطع مرجع التدقيق النحوي XML BNF كيفية قراءة BNF Grammar ويتضمن قواعد BNF من أجل XML 1.0 وأمثلة لمنتجات XML 1.0. يوضح مقطع مرجع قيود الصياغة الجيدة معنى قيد الصياغة الجيدة كما يسرد المنتجات المرفقة مع قيود الصياغة الجيدة. يشرح المقطع الخاص بمرجع قيود التحقق من الصحة ما هو قيد التحقق من الصحة كما يسرد ويشرح جميع قيود التحقق من الصحة في XML 1.0 Standard.

التدقيق النحوي XML BNF

وفقاً لمواصفات XML 1.0، يعتبر مستند XML مصاغ بشكل جيد إذا:

١- تلاءم مع مستند الإنتاج المعنون.

٢- يتوافق مع جميع قيود الصياغة الجيدة الموجودة في المواصفات.

٣- إذا كانت كل الكينونات الموزعة التي يشار إليها مباشرة أو بطريقة غير مباشرة في المستند مصاغة بشكل جيد.

يتم تصميم هذا المقطع ليساعدك على فهم أول هذه الشروط وتحديد ما إذا كانت هذه المستندات تتوافق مع هذا الشرط.

قراءة التدقيق النحوي BNF

يعتبر BNF هو اختصار Backus-Naur-Form ونتيجة لنظرية الترجمة. يقوم التدقيق النحوي BNF بتعريف البرنامج الصحيح والخطأ من ناحية بناء الجملة أو المستند الصحيح في بناء جمله "إعرابياً" في حالة XML. من المحتمل مقارنة مستند بالتدقيق النحوي BNF وتحديد ما إذا كان يتوافق مع شروط هذا التدقيق النحوي أم لا. لا يوجد حالات خط حدود. يحتوي التدقيق النحوي BNF المكتوب بطريقة صحيحة على ميزة عدم إتاحة أي مساحة للتفسير والتي من المفترض أن تتضح لأي فرد يتعامل مع مستندات HTML التي يتم عرضها في مستعرض واحد فقط.



فنياً، يستخدم XML التدقيق النحـــوي Extended-Backus-Naur-Form الـــذي يضيف بعض النقاط التي لا توجد في التدقيق النحوي BNF بـــالمترجم الموجـــه التقليدي.

تعتبر صحة بناء الجملة شرطاً ضرورياً ولكن غير كافي لمستندات XML. قد يلتحق مســـتند بالتدقيق النحوي BNF ومع ذلك، يفشل في أن يكون مصاغ بشكل جيد أو أن يكـــون صحيحـــاً. ولكي يتم صياغة مستند بشكل جيد، يجب أن يتوافق مع كل قيود الصياغة الجيـــدة لمواصفـــات XML 1.0. تعتبر الصياغة الجيدة هي أدنى مستوى يمكن أن يحققه المستند لكي يتـــم توزيعـــه. يجب أن يتوافق المستند مع كل قيود الصحة لمواصفات XML 1.0 حتى يكون المستند صحيحـــاً. يتم مناقشة قيود الصياغة الجيدة والتحقق من الصحة في المقطعين التاليين لهذا الملحق بالترتيب.

أجزاء التدقيق النحوي BNF (BNF Grammar Parts)

يحتوي التدقيق النحوي BNF على ثلاثة أجزاء:

١- مجموعة من السلاسل الحرفية تسمى نهايات. على سبيل المثال، يطلق على كـــل مـــن #REQUIRED, <, >, </, CDATA, و<!ENTITY نهايات يتـــم اســـتخدامها فـــي مواصفات XML 1.0.

٢- مجموعة من المحطات غير الطرفية "non-terminals" ليتم استبدالها بمحطات طرفيـــة "terminals".

٣- قائمة من المنتجات أو القواعد التي تضع المحطات غير الطرفية فـــي تسلســـل معيـــن للمحطات الطرفية وأخرى غير الطرفية بما في ذلك واحدة يتم تعريفها خصيصاً كمنتـــج البداية أو المستند.

إذا لم تكن مؤلف نظرية، قد يتم كتابة هذه القائمة في لغة Etruscan. ستتضح الأشياء بشـــكل أفضل من خلال عرض مثال بسيط قبل التعمق في تعقيدات التدقيق النحوي XML 1.0.

اعتبر السلاسل المكونة من الأعداد الصحيحة غير السالبة المضافة إلى أو المطروحـــة مـــن بعضها، مثل الآتي:

```
9+8+1+2+3
8-1-2-4-5
9+8-9-0+5+3
4
4+3
```

لاحظ أن بعض الأشياء غير موجودة في القائمة ولا نرغب إدراجها بالتدقيق النحوي وهـــي كما يلي:

- أي حرف فيما عدا الأرقام من 0 إلى 9 وعلامتي الزائد والناقص.
- المسافة البيضاء.
- سلسلة تبدأ بكل من + أو -.
- أرقام أصغر من 0 أو أكبر 9.
- السلسلة الفارغة.

فيما يلي التدقيق النحوي BNF الذي يقوم بتعريف السلاسل المطلوبة فقط:

```
[1] string ::= digit
[2] digit  ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7'
               | '8' | '9'
[3] string ::= string '+' digit
[3] string ::= string '-' digit
```

افترض تحديد ما إذا كانت هذه السلسلة "٩+٣ ٢" ترضي هذا التدقيق النحوي. يتم بدء بحـث المنتج الأول مما يعني أن السلسلة هي رقم محطة غـــير طرفيـــة. لذلـــك يتـــم الانتقـــال إلـــى Production [2] الذي يعرف الرقم. يعتبر رقم ٩ بالفعل هو أحد المحطات الطرفية التي يتـــم سردها كرقم وبالتالي تعتبر السلسلة ٩ هي سلسلة حقيقية. يفترض Production [3] أن السلسلة المتبوعة بعلامة الجمع ورقم آخر هي أيضاً سلسلة. لذلك تلبي ٩+٣ احتياجات التدقيق النحـــوي بالإضافة إلى اعتبارها سلسلة. يفترض Production [4] أن السلسلة المتبوعة بعلامة النـــاقص ورقم آخر هي سلسلة حقيقية ولذلك تعتبر ٩+٣+٢ هي سلسلة حقيقية وتلبي احتياجـــات التدقيـــق النحوي.

لاحظ الآن السلسلة "١+٩". باستخدام Production [1] من المفترض أن تبدأ السلسلة برقـم. لا تبدأ هذه السلسلة برقم ولذلك غير مسموح بها.

يعتبر التدقيق النحوي XML 1.0 أكبر وأكثر تعقيداً عن التدقيق النحوي البسيط. يسرد المقطع التالي منتجاته التي يبلغ عددها ٨٩ منتج. يوضح المقطع التالي كل منتج بالتفصيل.

رموز BNF

يتم استخدام الرموز الأساسية التالية في تدقيق EBNF النحوي الخاص ببرنامج XML على الجانب الأيمن للمنتجات:

#Xn	يعتبر N هو عدد ست عشري صحيح ويعرف #xN كحرف Unicode بالعدد N.
[a-zA-Z]	يطابق أي حرف في النطاق المحدد.
[#xN-#xN]	يطابق أي حرف في النطاق المحدد حيث تعتبر N هي قيمة ست عشرية لحرف Unicode.
[^a-z]	يطابق أي حرف غير موجود في النطاق المحدد.
[^#xN-#xN]	يطابق أي حرف غير موجود في النطاق المحدد حيث تعتبر N هي قيمة ست عشرية للحرف Unicode.
[^abc]	يطابق أي حرف غير موجود في القائمة.
[^#xN#xN#xN]	يطابق أي حرف لا توجد قيمته في القائمة.
String	يطابق السلسلة الحرفية الموجودة في علامات اقتباس فردية.
String	يطابق السلسلة الحرفية الموجودة في علامات اقتباس مزدوجة.

قد يتم تجميع هذه الأمثلة التسع الأساسية لتطابق التعبيرات الأكثر تعقيداً:

(contents)	يتم معاملة محتويات الأقواس كوحدة.
A?	تطابق zero أو أي تواجد لحرف A.
A B	يطابق A متبوع بحرف B.
A | B	يطابق A أو B ولكن ليس كليهما.
A – B	يطابق أي سلسلة تطابق A ولا تطابق B.
A+	يطابق تواجد واحد أو أكثر للحرف A.
A*	يطابق zero أو أكثر من تواجد لحرف A.

تستخدم تعيينات XML أيضاً ثلاثة نماذج لن تواجهها في تعيينات XML غير المرتبطة:

/* text of comment */	هو تعليق يتم تجاهل أي نص بداخله.
[WFC: name]	يقوم ذلك بتسمية قيد صياغة جيدة مرفق بهذا المنتج الذي يجب أن تتوافق معه المستندات حتى يتم تصنيفه كجيد الصياغة. توجد قيود الصياغة الجيدة في التعيين ولكن لا يتم تغليفها في تدقيق BNF النحوي.

[VC: name]

يقوم ذلك بتسمية قيد التحقق من الصحة المرفق مع هذا المنتج الذي يجب أن تتوافق معه المستندات حتى يتم تصنيفه كجيد الصياغة "التكوين". توجد قيود التحقق من الصحة في التعيين ولكن لا يتم تغليفها في تدقيق BNF النحوي.

قواعد BNF للغة XML 1.0

يتاح التدقيق النحوي الكامل BNF الخاص بلغة XML في مواصفات XML 1.0 التي ستجدها فــي ملحق ب من هذا الكتاب. ومع ذلك، إذا أردت مطابقة الترميز في مقابل نتاج التدقيق النحوي، قد يكون غير ملائماً استعراض الصفحات بحثاً عن القواعد الضرورية. لذلك يتم نسخ قواعــد BNF من أجل XML1.0 هنا.

المستند

```
[1] document ::= prolog element Misc*
```

نطاق الأحرف

```
[2] Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD]
             | [#x10000-#x10FFFF]
```

مسافة بيضاء

```
[3] S ::= (#x20 | #x9 | #xD | #xA)+
```

أسماء ورموز

```
[4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':'
                 | CombiningChar | Extender
[5] Name     ::= (Letter | '_' | ':') (NameChar)*
[6] Names    ::= Name (S Name)*
[7] Nmtoken  ::= (NameChar)+
[8] Nmtokens ::= Nmtoken (S Nmtoken)*
```

أحرف

```
[9]  EntityValue   ::= ' " ' ([^%&"] | PEReference | Reference)*
                   ' " ' | " ' " ([^%&'] | PEReference
                   | Reference)* " ' "
[10] AttValue      ::= ' " ' ([^<&"] | Reference)* ' " '
                   | "'" ([^<&'] | Reference)* "'"
[11] SystemLiteral ::= ('"' [^"]* '"') | ("'" [^']* "'")
[12] PubidLiteral  ::= '"' PubidChar* '"'
                   | "'" (PubidChar – "'")* "'"
[13] PubidChar     ::= #x20 | #xD | #xA | [a-zA-Z0-9]
                   | [-'()+,./:=?;!*#@$_%]
```

بيانات الأحرف

```
[14] CharData ::= [^<&]* - ([^<&]* ']]>' [^<&]*)
```

التعليقات

```
[15] Comment ::= '<!–' ((Char – '-')
             | ('–' (Char – '–')))* '–>'
```

تعليمات التشغيل

```
[16] PI ::= '<?' PITarget
            (S (Char* - (Char* '?>' Char*)))? '?>'
[17] PITarget ::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))
```

مقاطع CDATA

```
[18] CDSect  ::= CDStart CData CDEnd
[19] CDStart ::= '<![CDATA['
[20] CData   ::= (Char* - (Char* ']]>' Char*))
[21] CDEnd   ::= ']]>'
```

مقدمة "prolog"

```
[22] prolog       ::= XMLDecl? Misc* (doctypedecl Misc*)?
[23] XMLDecl      ::= '<?xml' VersionInfo EncodingDecl? SDDecl?
                      S? '?>'
[24] VersionInfo ::= S 'version' Eq (' VersionNum '
                      | " VersionNum ")
[25] Eq           ::= S? '=' S?
[26] VersionNum::= ([a-zA-Z0-9_.:] | '-')+
[27] Misc         ::= Comment | PI |  S
```

تعريف نوع المستند

```
[28] doctypedecl  ::= '<!DOCTYPE' S Name (S ExternalID)?
                      S? ('[' (markupdecl | PEReference
                      | S)* ']' S?)? '>'
                      [ VC: Root Element Type ]
[29] markupdecl  ::= elementdecl | AttlistDecl
                      | EntityDecl | NotationDecl | PI
                      | Comment
                      [ VC: Proper Declaration/PE Nesting ]
                      [ WFC: PEs in Internal Subset ]
```

المجموعة الجزئية الخارجية

```
[30] extSubset       ::=   TextDecl? extSubsetDecl
[31] extSubsetDecl ::=   ( markupdecl | conditionalSect | PEReference |
S )*
```

إعلان مستند مستقل

```
[32] SDDecl ::= S 'standalone' Eq (("'" ('yes' | 'no')
               "'") | ('"' ('yes' | 'no') '"'))
               [ VC: Standalone Document Declaration ]
```

تعريف اللغة

```
[33] LanguageID ::= Langcode ('-' Subcode)*
[34] Langcode    ::= ISO639Code | IanaCode | UserCode
[35] ISO639Code ::= ([a-z] | [A-Z]) ([a-z] | [A-Z])
[36] IanaCode    ::= ('I' | 'I') '-' ([a-z] | [A-Z])+
[37] UserCode    ::= ('x' | 'X') '-' ([a-z] | [A-Z])+
[38] Subcode     ::= ([a-z] | [A-Z])+
```

العنصر

```
[39] element ::= EmptyElemTag | STag content ETag
                 [ WFC: Element Type Match ]
                 [ VC: Element Valid ]
```

علامة البدء

```
[40] STag      ::= '<' Name (S Attribute)* S? '>'
                 [ WFC: Unique Att Spec ]
[41] Attribute ::= Name Eq AttValue
                 [ VC: Attribute Value Type ]
                 [ WFC: No External Entity References ]
                 [ WFC: No < in Attribute Values ]
```

علامة الانتهاء

```
[42] ETag ::= '</' Name S? '>'
```

محتوى العناصر

```
[43] content ::= (element | CharData | Reference | CDSect
                 | PI | Comment)*
```

علامات العناصر الفارغة

```
[44] EmptyElemTag ::= '<' Name (S Attribute)* S? '/>'
                      [ WFC: Unique Att Spec ]
```

إعلان نوع العنصر

```
[45] elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'
                     [ VC: Unique Element Type Declaration ]
[46] contentspec ::= 'EMPTY' | 'ANY' | Mixed | children
```

أنواع محتوى العنصر

```
[47] children ::= (choice | seq) ('?' | '*' | '+')?
[48] cp       ::= (Name | choice | seq) ('?' | '*' | '+')?
[49] choice   ::= '(' S? cp ( S? '|' S? cp )* S? ')'
                  [ VC: Proper Group/PE Nesting ]
[50] seq      ::=  '(' S? cp ( S? ',' S? cp )* S? ')'
                  [ VC: Proper Group/PE Nesting ]
```

إعلان المحتوى المختلط

```
[51] Mixed ::= '(' S? '#PCDATA' (S? '|' S? Name)* S? ')*'
             | '(' S? '#PCDATA' S? ')'
             [ VC: Proper Group/PE Nesting ]
             [ VC: No Duplicate Types ]
```

إعلان قائمة السمات

```
[52] AttlistDecl ::= '<!ATTLIST' S Name AttDef* S? '>'
[53] AttDef      ::= S Name S AttType S DefaultDecl
```

أنواع السمات

```
[54] AttType        ::= StringType | TokenizedType | EnumeratedType
[55] StringType     ::= 'CDATA'
[56] TokenizedType  ::= 'ID' | 'IDREF' | 'IDREFS' | 'ENTITY'
                      | 'ENTITIES' | 'NMTOKEN' | 'NMTOKENS'
                      [ VC: ID ]
                      [ VC: One ID per Element Type ]
                      [ VC: ID Attribute Default ]
```

[VC: IDREF]
[VC: Entity Name]
[VC: Name Token]

أنواع السمات المتعددة

```
[57] EnumeratedType ::= NotationType | Enumeration
[58] NotationType    ::= 'NOTATION' S '(' S? Name (S? '|' S?
                          Name)* S? ')'
                        [ VC: Notation Attributes ]
[59] Enumeration     ::= '(' S? Nmtoken (S? '|' S? Nmtoken)*
                          S? ')'
                        [ VC: Enumeration ]
```

افتراضات السمات

```
[60] DefaultDecl ::= '#REQUIRED' | '#IMPLIED'
                   | (('#FIXED' S)? AttValue)
                   [ VC: Required Attribute ]
                   [ VC: Attribute Default Legal ]
                   [ WFC: No < in Attribute Values ]
                   [ VC: Fixed Attribute Default ]
```

المقطع الشرطي

```
[61] conditionalSect     ::=  includeSect | ignoreSect
[62] includeSect         ::= '<![' S? 'INCLUDE' S? '['
                              extSubsetDecl ']]>'
[63] ignoreSect          ::= '<![' S? 'IGNORE' S? '['
                              ignoreSectContents* ']]>'
[64] ignoreSectContents ::= Ignore ('<![' ignoreSectContents
                              ']]>' Ignore)*
[65] Ignore              ::= Char* - (Char* ('<![' | ']]>') Char*)
```

مرجع الحرف

[66] CharRef ::= '&#' [0 -9]+ ';' | '&#x' [0 -9a –fA -F]+ ';'
[WFC: Legal Character]

مرجع الكينونة

[67] Reference ::= EntityRef | CharRef
[68] EntityRef ::= '&' Name ';'
[WFC: Entity Declared]
[VC: Entity Declared]
[WFC: Parsed Entity]
[WFC: No Recursion]
[69] PEReference ::= '%' Name ';'
[VC: Entity Declared]
[WFC: No Recursion]
[WFC: In DTD]

إعلان الكينونة

[70] EntityDecl ::= GEDecl | PEDecl
[71] GEDecl ::= '<!ENTITY' S Name S EntityDef S? '>'
[72] PEDecl ::= '<!ENTITY' S '%' S Name S PEDef S? '>'
[73] EntityDef ::= EntityValue | (ExternalID NDataDecl?)
[74] PEDef ::= EntityValue | ExternalID

إعلان الكينونة الخارجي

[75] ExternalID ::= 'SYSTEM' S SystemLiteral
| 'PUBLIC' S PubidLiteral S SystemLiteral
[76] NDataDecl ::= S 'NDATA' S Name
[VC: Notation Declared]

إعلان النص

[77] TextDecl ::= '<?xml' VersionInfo? EncodingDecl S? '?>'

كينونة موزعة خارجية جيدة الصياغة "Well-formed External Parsed Entity"

```
[78] extParsedEnt ::= TextDecl? content
[79] extPE        ::= TextDecl? extSubsetDecl
```

إعلانات محولة إلى رموز

```
[80] EncodingDecl ::= S 'encoding' Eq ('"' EncName '"'
                      | "'" EncName "'" )
[81] EncName      ::= [A-Za-z] ([A-Za-z0-9._] | '-')*
```

إعلانات مجموعة رموز

```
[82] NotationDecl ::= '<!NOTATION' S Name S (ExternalID
                      | PublicID) S? '>'
[83] PublicID     ::= 'PUBLIC' S PubidLiteral
```

الأحرف

```
[84] Letter    ::= BaseChar | Ideographic
[85] BaseChar ::= [#x0041-#x005A] | [#x0061-#x007A]
                 | [#x00C0-#x00D6] | [#x00D8-#x00F6]
                 | [#x00F8-#x00FF] | [#x0100-#x0131]
                 | [#x0134-#x013E] | [#x0141-#x0148]
                 | [#x014A-#x017E] | [#x0180-#x01C3]
                 | [#x01CD-#x01F0] | [#x01F4-#x01F5]
                 | [#x01FA-#x0217] | [#x0250-#x02A8]
                 | [#x02BB-#x02C1] | #x0386 | [#x0388-#x038A]
                 | #x038C | [#x038E-#x03A1] | [#x03A3-#x03CE]
                 | [#x03D0-#x03D6] | #x03DA | #x03DC | #x03DE
                 | #x03E0 | [#x03E2-#x03F3] | [#x0401-#x040C]
                 | [#x040E-#x044F] | [#x0451-#x045C]
                 | [#x045E-#x0481] | [#x0490-#x04C4]
                 | [#x04C7-#x04C8] | [#x04CB-#x04CC]
```

```
| [#x04D0-#x04EB] | [#x04EE-#x04F5]
| [#x04F8-#x04F9] | [#x0531-#x0556] | #x0559
| [#x0561-#x0586] | [#x05D0-#x05EA]
| [#x05F0-#x05F2] | [#x0621-#x063A]
| [#x0641-#x064A] | [#x0671-#x06B7]
| [#x06BA-#x06BE] | [#x06C0-#x06CE]
| [#x06D0-#x06D3] | #x06D5 | [#x06E5-#x06E6]
| [#x0905-#x0939] | #x093D | [#x0958-#x0961]
| [#x0985-#x098C] | [#x098F-#x0990]
| [#x0993-#x09A8] | [#x09AA-#x09B0]
| #x09B2 | [#x09B6-#x09B9] | [#x09DC-#x09DD]
| [#x09DF-#x09E1] | [#x09F0-#x09F1]
| [#x0A05-#x0A0A] | [#x0A0F-#x0A10]
| [#x0A13-#x0A28] | [#x0A2A-#x0A30]
| [#x0A32-#x0A33] | [#x0A35-#x0A36]
| [#x0A38-#x0A39] | [#x0A59-#x0A5C]
| #x0A5E | [#x0A72-#x0A74] | [#x0A85-#x0A8B]
| #x0A8D | [#x0A8F-#x0A91] | [#x0A93-#x0AA8]
| [#x0AAA-#x0AB0] | [#x0AB2-#x0AB3]
| [#x0AB5-#x0AB9] | #x0ABD | #x0AE0
| [#x0B05-#x0B0C] | [#x0B0F-#x0B10]
| [#x0B13-#x0B28] | [#x0B2A-#x0B30]
| [#x0B32-#x0B33] | [#x0B36-#x0B39]
| #x0B3D | [#x0B5C-#x0B5D] | [#x0B5F-#x0B61]
| [#x0B85-#x0B8A] | [#x0B8E-#x0B90]
| [#x0B92-#x0B95] | [#x0B99-#x0B9A] | #x0B9C
| [#x0B9E-#x0B9F] | [#x0BA3-#x0BA4]
| [#x0BA8-#x0BAA] | [#x0BAE-#x0BB5]
| [#x0BB7-#x0BB9] | [#x0C05-#x0C0C]
| [#x0C0E-#x0C10] | [#x0C12-#x0C28]
| [#x0C2A-#x0C33] | [#x0C35-#x0C39]
| [#x0C60-#x0C61] | [#x0C85-#x0C8C]
| [#x0C8E-#x0C90] | [#x0C92-#x0CA8]
| [#x0CAA-#x0CB3] | [#x0CB5-#x0CB9] | #x0CDE
```

```
                    | [#x0CE0-#x0CE1] | [#x0D05-#x0D0C]
                    | [#x0D0E-#x0D10] | [#x0D12-#x0D28]
                    | [#x0D2A-#x0D39] | [#x0D60-#x0D61]
                    | [#x0E01-#x0E2E] | #x0E30 | [#x0E32-#x0E33]
                    | [#x0E40-#x0E45] | [#x0E81-#x0E82] | #x0E84
                    | [#x0E87-#x0E88] | #x0E8A | #x0E8D
                    | [#x0E94-#x0E97] | [#x0E99-#x0E9F]
                    | [#x0EA1-#x0EA3] | #x0EA5 | #x0EA7
                    | [#x0EAA-#x0EAB] | [#x0EAD-#x0EAE] | #x0EB0
                    | [#x0EB2-#x0EB3] | #x0EBD | [#x0EC0-#x0EC4]
                    | [#x0F40-#x0F47] | [#x0F49-#x0F69]
                    | [#x10A0-#x10C5] | [#x10D0-#x10F6] | #x1100
                    | [#x1102-#x1103] | [#x1105-#x1107] | #x1109
                    | [#x110B-#x110C] | [#x110E-#x1112] | #x113C
                    | #x113E | #x1140 | #x114C | #x114E | #x1150
                    | [#x1154-#x1155] | #x1159 | [#x115F-#x1161]
                    | #x1163 | #x1165 | #x1167 | #x1169
                    | [#x116D-#x116E] | [#x1172-#x1173] | #x1175
                    | #x119E | #x11A8 | #x11AB | [#x11AE-#x11AF]
                    | [#x11B7-#x11B8] | #x11BA | [#x11BC-#x11C2]
                    | #x11EB | #x11F0 | #x11F9 | [#x1E00-#x1E9B]
                    | [#x1EA0-#x1EF9] | [#x1F00-#x1F15]
                    | [#x1F18-#x1F1D] | [#x1F20-#x1F45]
                    | [#x1F48-#x1F4D] | [#x1F50-#x1F57] | #x1F59
                    | #x1F5B | #x1F5D | [#x1F5F-#x1F7D]
                    | [#x1F80-#x1FB4] | [#x1FB6-#x1FBC] | #x1FBE
                    | [#x1FC2-#x1FC4] | [#x1FC6-#x1FCC]
                    | [#x1FD0-#x1FD3] | [#x1FD6-#x1FDB]
                    | [#x1FE0-#x1FEC] | [#x1FF2-#x1FF4]
                    | [#x1FF6-#x1FFC] | #x2126 | [#x212A-#x212B]
                    | #x212E | [#x2180-#x2182] | [#x3041-#x3094]
                    | [#x30A1-#x30FA] | [#x3105-#x312C]
                    | [#xAC00-#xD7A3]
[86] Ideographic    ::= [#x4E00-#x9FA5] | #x3007
```

```
                         | [#x3021-#x3029]
[87] CombiningChar ::= [#x0300-#x0345] | [#x0360-#x0361]
                         | [#x0483-#x0486] | [#x0591-#x05A1]
                         | [#x05A3-#x05B9] | [#x05BB-#x05BD]
                         | #x05BF | [#x05C1-#x05C2] | #x05C4
                         | [#x064B-#x0652] | #x0670
                         | [#x06D6-#x06DC] | [#x06DD-#x06DF]
                         | [#x06E0-#x06E4] | [#x06E7-#x06E8]
                         | [#x06EA-#x06ED] | [#x0901-#x0903]
                         | #x093C | [#x093E-#x094C] | #x094D
                         | [#x0951-#x0954] | [#x0962-#x0963]
                         | [#x0981-#x0983] | #x09BC | #x09BE
                         | #x09BF | [#x09C0-#x09C4]
                         | [#x09C7-#x09C8] | [#x09CB-#x09CD]
                         | #x09D7 | [#x09E2-#x09E3] | #x0A02
                         | #x0A3C | #x0A3E | #x0A3F
                         | [#x0A40-#x0A42] | [#x0A47-#x0A48]
                         | [#x0A4B-#x0A4D] | [#x0A70-#x0A71]
                         | [#x0A81-#x0A83] | #x0ABC
                         | [#x0ABE-#x0AC5] | [#x0AC7-#x0AC9]
                         | [#x0ACB-#x0ACD] | [#x0B01-#x0B03]
                         | #x0B3C | [#x0B3E-#x0B43]
                         | [#x0B47-#x0B48] | [#x0B4B-#x0B4D]
                         | [#x0B56-#x0B57] | [#x0B82-#x0B83]
                         | [#x0BBE-#x0BC2] | [#x0BC6-#x0BC8]
                         | [#x0BCA-#x0BCD] | #x0BD7
                         | [#x0C01-#x0C03] | [#x0C3E-#x0C44]
                         | [#x0C46-#x0C48] | [#x0C4A-#x0C4D]
                         | [#x0C55-#x0C56] | [#x0C82-#x0C83]
                         | [#x0CBE-#x0CC4] | [#x0CC6-#x0CC8]
                         | [#x0CCA-#x0CCD] | [#x0CD5-#x0CD6]
                         | [#x0D02-#x0D03] | [#x0D3E-#x0D43]
                         | [#x0D46-#x0D48] | [#x0D4A-#x0D4D]
                         | #x0D57 | #x0E31 | [#x0E34-#x0E3A]
```

```
                   | [#x0E47-#x0E4E] | #x0EB1
                   | [#x0EB4-#x0EB9] | [#x0EBB-#x0EBC]
                   | [#x0EC8-#x0ECD] | [#x0F18-#x0F19]
                   | #x0F35 | #x0F37 | #x0F39 | #x0F3E
                   | #x0F3F | [#x0F71-#x0F84]
                   | [#x0F86-#x0F8B] | [#x0F90-#x0F95]
                   | #x0F97 | [#x0F99-#x0FAD]
                   | [#x0FB1-#x0FB7] | #x0FB9
                   | [#x20D0-#x20DC] | #x20E1
                   | [#x302A-#x302F] | #x3099 | #x309A
[88] Digit ::=       [#x0030-#x0039] | [#x0660-#x0669]
                   | [#x06F0-#x06F9] | [#x0966-#x096F]
                   | [#x09E6-#x09EF] | [#x0A66-#x0A6F]
                   | [#x0AE6-#x0AEF] | [#x0B66-#x0B6F]
                   | [#x0BE7-#x0BEF] | [#x0C66-#x0C6F]
                   | [#x0CE6-#x0CEF] | [#x0D66-#x0D6F]
                   | [#x0E50-#x0E59] | [#x0ED0-#x0ED9]
                   | [#x0F20-#x0F29]
[89] Extender ::=    #x00B7 | #x02D0 | #x02D1 | #x0387
                   | #x0640 | #x0E46 | #x0EC6 | #x3005
                   | [#x3031-#x3035] | [#x309D-#x309E]
                   | [#x30FC-#x30FE]
```

أمثلة عن منتجات XML 1.0

يوضح هذا المقطع بعض الأمثلة عن المنتجات لإعطاء فكرة جيدة عن معنى كل منتج.

المستند

```
[1] document ::= prolog element Misc*
```

توضح هذه القاعدة أن مستند XML يتكون من مقدمة (Production [22]) متبوعة بعنصر جذري واحد (Production [39]) متبوع بأي عدد من العناصر المتنوعة (Production [27]) أي أن البنية النموذجية تبدو كما يلي:

```
<?xml version="1.0"?>
```

```
<!- a DTD might go here ->
<ROOT_ELEMENT>
 Content
</ROOT_ELEMENT>
<!- comments can go here ->
<?Reader, processing instructions can also go here?>
```

لا يمكن أن يتبع أي شيء إغلاق العنصر الجذري.

يلغي [1] Production المستندات التي تحتوي على أكثر من عنصر يعتبر جذر. على سبيل المثال،

```
<?xml version="1.0"?>
<ELEMENT1>
 Content
</ELEMENT1>
<ELEMENT2>
 Content
</ELEMENT2>
<ELEMENT1>
 Content
</ELEMENT1>
```

نطاق الحرفي

[2] Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]

يقوم [2] Production بتعريف المجموعة الجزئية لأحرف Unicode التي قد تظهر مستند XML. تعتبر العناصر الأساسية الهامة هنا هي الأحرف التي لم يتم تضمينها وبالأخص، أحرف عناصر تحكم ASCII غير المطبوعة والتي يعتبر الشائع منها هو الجرس وعلامة التبويب العمودية وملء النموذج "formfeed"و block surrogates من #xD800 إلى #xDFFF و#xFFFE غير الحرفي. لا تعتبر أحرف عناصر التحكم في XML ضرورية مما قد يسبب مشاكل في الملفات المعروضة على المحطات الطرفية القديمة أو التي تمر في خادم المحطات الطرفية القديمة والبرامج.

سيتم استخدام "surrogates block" لتوسيع Unicode لدعم أكثر من مليون حرف مختلف ومع ذلك لم يتم تعريف أياً من هذه الإضافات التي يبلغ عددها مليون ولم يتاح لموزعي XML دعمها.

لم يتم تعريف xFFFE# الذي لا يعتبر حرف فـــي Unicode. ويجـــب أن يشـــير مظـــهره وبالأخص في بداية المستند إلى قراءتك للمستند الذي يحتوي على ترتيب البايت الخاطئ والـــذي يعتبر little endian بدلاً من big endian والعكس صحيح.

المسافة البيضاء

[3] S ::= (#x20 | #x9 | #xD | #xA)+

يقوم [3] Production بتعريف المسافة البيضاء كتشغيل لحـــرف مســـافة واحـــد أو أكـــثر (#x20) وعلامة التبويب الأفقية (#x9) وحرف الإرجاع (#xD) وملء الســـطر "linefeed" (#xA). يتم معاملة عشرين حرف من تلك الموجودة في صف كحرف واحد بـــالضبط بســـبب علامة +. يقوم [2] production بمنع أحرف المسافة البيضاء ASCII الأخرى مثـــل علامـــة التبويب العمودية (#xB). لا تعتبر أحرف المسافة البيضاء non-ASCII وUnicode الأخـــرى مثل المسافة غير المنقسمة (#A0) مسافة بيضاء لأغراض XML.

أسماء ورموز

[4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':' | CombiningChar | Extender

يقوم [4] Production الحروف التي قد تظهر في اسم XML وقد تحتوي أسماء XML علـــى أحرف وأرقام ونقاط وواصلات وتسطير أسفل الأسطر وعلامـــات النقطتيـــن وأحـــرف ربـــط (Production [87]) و"extenders" (Production [89]).

[5] Name ::= (Letter | '_' | ':') (NameChar)*

يوضح [5] Production أن اسم XML يجب أن يبدأ بحرف أو بتسطير أســـفل الســـطر أو بعلامة نقطتين. قد لا يبدأ برقم أو بنقطة أو بواصلة. كما تحتوي الأحرف المتتالية في اسم XML على أي حرف اسم XML (Production [4]) بما في ذلك الأرقام والنقاط والواصلات. وفيمـــا يلي أسماء XML المصرح بها:

airplane
text.encoding
r
SEAT
Pilot
Pilot1
OscarWilde

BOOK_TITLE
:TITLE
_8ball
«inoyz

وفيما يلي أسماء XML غير المصرح بها:

air plane
.encoding
-r
Wilde,Oscar
BOOK TITLE
8ball
AHA!

[6] Names ::= Name (S Name)*

يقوم Production [6] بتعريف مجموعة من الأسماء كاسم أو أكثر لأسماء XML (Production [5]) يتم فصلها بواسطة مسافة بيضاء. وفيما يلي المجموعة الصحيحة لأسماء XML:

BOOK AUTHOR TITLE PAGE EDITOR CHAPTER

لا يعتبر ما يلي مجموعة صحيحة لأسماء XML:

BOOK, AUTHOR, TITLE, PAGE, EDITOR, CHAPTER

[7] Nmtoken ::= (NameChar)+

يقوم Production [7] بتعريف رمز اسم كأي تسلسل لحرف أو أكثر من أحرف الاسم وبخلاف اسم XML، لا يحتوي رمز الاسم على أي قيود خاصة بطول الحرف الأول الذي يعتبر صحيحاً ما دام حرف الاسم صحيحاً (Production [4]) أي أن رموز أسماء XML قد تبدأ برقم أو نقطة أو واصلة بينما لا يبدأ بها اسم XML. تعتبر كل أسماء XML هي رموز أسماء XML صحيحة، ولكن ليست كل رموز الأسماء الصحيحة أسماء XML صحيحة.

فيما يلي رموز الأسماء المصرح بها:

airplane
text.encoding
r
SEAT
Pilot
Pilot1
OscarWilde

BOOK_TITLE
:TITLE
_8ball
«inoyz
.encoding
-r
8ball

وفيما يلي رموز الأسماء غير المصرح بها:

air plane
Wilde,Oscar
BOOK TITLE
AHA!

[8] Nmtokens ::= Nmtoken (S Nmtoken)*

يوضح [8] Production أن مجموعة من رموز الأسماء هي أحد رمـــوز أســماء XML أو أكثر ([7] Production) مفصولة بمسافة بيضاء. يعتبر ذلك مجموعة صحيحة من رموز أسماء XML:

1POTATO 2POTATO 3POTATO 4POTATO

لا يعتبر ذلك مجموعة صحيحة من رموز أسماء XML:

1POTATO, 2POTATO, 3POTATO, 4POTATO

أحرف

[9] EntityValue ::= '"' ([^%&"] | PEReference | Reference)* '"' | "'" ([^%&'] | PEReference | Reference)* "'"

يقوم [9] Production بتعريف قيمة كينونة كأي سلسلة من الأحـــرف المحاطـــة بعلامـــات اقتباس مزدوجة أو بعلامة اقتباس واحدة ما عدا % و& وحرف علامة الاقتباس "ســواء كــانت فردية أو مزدوجة" يتم استخدامها لتحديد السلسلة. قد يتم استخدام % و& ســواء كــانوا بدايــة مرجع كينونة معامل ([69] Production) أو مرجع كينونـــة عـــام ([67] Production) أو مرجع حرف. إذا أردت تضمين % و& في قيم الكينونة، يمكنك الهروب منها باستخدام مراجـــع الأحرف % و& بالترتيب.

فيما يلي قيم الكينونة المصرح بها:

"This is an entity value"
'This is an entity value'
"75% off"

"Ben & Jerry's New York Super Fudge Chunk Ice Cream"

وفيما يلي قيم الكينونة غير الصحيحة:

This is an entity value'
'This is an entity value"
"75% off"
"Ben & Jerry's New York Super Fudge Chunk Ice Cream"
'Ben & Jerry's New York Su

[10] AttValue ::= '"' ([^<&"] | Reference)* '"' | "'" ([^<&'] | Reference)* "'"

يوضح Production [10] أن قيمة السمة قد تتكون من أي أحرف ما عدا < و& و' محاطة في علامات اقتباس مزدوجة أو أي أحرف محاطة في علامات اقتباس فردية. ومع ذلك قد تظهر & إذا تم استخدامها كبداية مرجع فقط (Production [67]) "سواء كانت عامة أو حرف".

فيما يلي قيم السمات الصحيحة:

"This is an attribute value"
'This is an attribute value'
'#FFCC33'
"75% off"
"Ben & Jerry's New York Super Fudge Chunk Ice Cream"
"i < j"

فيما يلي قيم السمات غير الصحيحة:

"This is an attribute value"
'This is an attribute value'
"Ben & Jerry's New York Super Fudge Chunk Ice Cream"
'Ben & Jerry's New York Super Fudge Chunk Ice Cream'
"i < j"

[11] SystemLiteral ::= ('"' [^"]* '"') | ("'" [^']* "'")

يقوم Production [11] بتعريف حرفية النظام كأي سلسلة نص لا تحتـــوي علـــى علامـــة اقتباس مزدوجة محاطة علامات اقتباس مزدوجة. وقد تكون حرفية النظام هي أي سلسلة نـــص لا يحتوي على علامة الاقتباس الفردية محاطة بعلامات اقتباس فردية. وفيما يلي أحـــرف نظـــام التدقيق النحوي:

"test"
" Hello there! "

' Hello
there!'
"Embedded markup is <OK/> in system literals"

فيما يلي أحرف نظام التدقيق غير النحوي:

" He said, "Get out of here!""
'bailey's Cove'

[12] PubidLiteral ::= '"' PubidChar* '"' | "'" (PubidChar – "'")* "'"

يوضح Production [12] أن رقم المعرف العام هو صفر أو أحرف معرفات عامة أكثر (Production [13]) محاطة بعلامات اقتباس مزدوجة أو بصفر أو بأحرف معرفات عامة أكثر ما عدا علامة الاقتباس الفردية المحاطة في علامات اقتباس فردية.

فيما يلي أرقام المعرفات العامة النحوية:

"IETF//NONSGML Media Type application/pdf//EN"
'-//IETF//NONSGML Media Type application/pdf//EN'
"-//W3C//DTD XHTML 1.0 Strict + Math//EN"

وفيما يلي أرقام المعرفات العامة غير النحوية:

"{-//IETF//NONSGML Media Type application/pdf//EN}"
"-//IETF//NONSGML Media Type application/¹__//GR}"

[13] PubidChar ::= #x20 | #xD | #xA | [a-zA-Z0-9] | [-'()+,./:=?;!*#@$_%]

يسرد Production [13] ASCII وحرف الإرجاع وملء السطر "linefeed" والأحرف من a إلى z ومن A إلى Z والأرقام من 0 إلى 9 وأحرف الترقيم و-ص%_$@#*!;?=:/.,+().

بيانات الأحرف

[14] CharData ::= [^<&]* - ([^<&]* ']]>'[^<&]*)

Production [14] بتعريف بيانات الحرف كأي عدد من الأحرف ما عدا > و&. بالإضافة إلى ذلك، قد لا يظهر <[[CDEnd string كجزء من بيانات الأحرف. وقد تحتوي بيانات الأحرف على بعض الأحرف الصفرية.

التعليقات

[15] Comment ::= '<!--' ((Char – '-') | ('-' (Char – '-')))* '-->'

Production [15] بتعريف تعليق كأي سلسلة نص محاطة بين علامتـي <! و > باسـتثناء الواصلة المزدوجة -. وفيما يلي التعليقات الصحيحة:

```
<!-Hello->
<!-Hello there!->
<!- Hello there! ->
<!- Hello
   there! ->
<!-<Hello/> <there/>!->
<!-<Hello/> </there>!->
```

لا يعتبر ذلك تعليق صحيح:

```
<!- Hello-there! ->
```

تعليمات التشغيل

[16] PI ::= '<?' PITarget (S (Char* - (Char* '?>' Char*)))? '?>'

يوضح Production [16] أن تعليمات التشغيل تبدأ بحرف <? متبوع باسم هدف تعليمـات التشغيل (Production [17]) ومتبوع بمسافة بيضاء متبوعة بأي عدد من الأحرف مـاعدا ?>. أخيراً، يغلق الحرف ?> تعليمات التشغيل.

فيما يلي تعليمات التشغيل الصحيحة:

```
<?gcc version="2.7.2" options="-O4"?>
<?Terri Do you think this is a good example?>
```

وفيما يلي تعليمات التشغيل غير الصحيحة:

```
<? I have to remember to fix this next part?>
<?Terri This is a good example!>
```

[17] PITarget ::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))

يوضح Production [17] أن هدف تعليمات التشـــغيل قـد يكـون أي اسـم للغـة XML (Production [5]) فيما عدا سلسلة XML "في أي مجموعة أحرف". وبالتالي، يعتبر كل ذلـك أهداف تعليمات تشغيل مقبولة:

```
gcc
acrobat
Acrobat
Joshua
Acrobat_301
```

xml-stylesheet
XML_Whizzy_Writer_2000

وفيما يلي أهداف تعليمات التشغيل غير المصرح بها:

xml
XML
xmL
CDATA Sections

[18] CDSect ::= CDStart CData CDEnd

يعلن [18] Production أن مقطع Cdata يتكون مـــن (Production [19]) CDStart و(Production [20]) CData و(Production [21]) CDEnd بنفس هذا الترتيب.

[19] CDStart ::= '<![CDATA['

يقوم [19] Production بتعريف CDStart كالسلسلة الحرفية <![CDATA[ولا شيئاً آخر.

[20] CData ::= (Char* - (Char* ']]>' Char*))

يوضح [20] Production أن مقطع Cdata قد يحتوي على أية أحرف فيما عـــدا سلســلة CDEnd.

[21] CDEnd ::= ']]>'

يقوم [21] Production بتعريف CDEnd كالسلسلة الحرفية]] ولا شيئاً آخر.

فيما يلي مقاطع CDATA الصحيحة:

```
<![CDATA[ The < character starts a tag in XML ]]>
<![CDATA[ CDATA sections begin with the literal <![CDATA[ ]]>
```

لا يعتبر ذلك مقطع CDATA صحيح:

```
<![CDATA[
 The three characters ]]> terminate a CDATA section
]]>
```

مقدمة "Prolog"

[22] prolog ::= XMLDecl? Misc* (doctypedecl Misc*)?

يوضح [22] Production أن المقدمة تتكون من إعلان XML اختياري متبـــوع بصفـــر أو بأكثر من عنصـــر متنـــوع (Production [27]) متبـــوع بـــإعلان نـــوع مســـتند اختيـــاري (Production [28]) متبوع بصفر أو بأكثر من عنصر متنوع. على سبيل المثال، يعتبر ما يلي

مقدمة صحيحة:

```
<?xml version="1.0"?>
```

يعتبر أيضاً ما يلي مقدمة صحيحة:

```
<?xml version="1.0" standalone="yes"?>
<?xml:stylesheet type='text/css" href="greeting.css"?>
<!DOCTYPE greeting [
  <!ELEMENT greeting (#PCDATA)>
]>
```

يعتبر أيضاً ما يلي مقدمة صحيحة:

```
<!-This strange document really doesn't have anything
   in its prolog! ->
```

لا يعتبر ما يلي مقدمة صحيحة نظراً لأن التعليق يسبق إعلان XML:

```
<!-This is from the example in Chapter 8 ->
<?xml version="1.0" standalone="yes"?>
<?xml:stylesheet type="text/css" href="greeting.css"?>
<!DOCTYPE greeting [
  <!ELEMENT greeting (#PCDATA)>
]>
```

[23] XMLDecl ::= '<?xml' VersionInfo EncodingDecl? SDDecl? S? '?>'

يقوم Production [23] بتعريف إعلان XML كسلسلة حرفيـــة <?xml متبوعـــة بسلســلة معلومات إصدار إجباري (Production [24])، متبوعة اختيارياً بإعلان محـــول إلـــى رمـــوز (Production [80]) وبإعلان مستند مستقل بذاتـــه (Production [32]) وبمســـافة بيضـــاء متبوعة بالسلسلة الحرفية ?>. وفيما يلي إعلانات XML الصحيحة:

```
<?xml version="1.0"?>
<?xml version="1.0" encoding="Big5"?>
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<?xml version="1.0" standalone="no"? >
<?xml version="1.0" encoding="ISO-8859-5"?>
```

لا يعتبر ما يلي إعلانات XML غير الصحيحة:

```
<?xml?>
<?xml encoding="Big5"?>
<?xml version="1.0" standalone="yes"? encoding="ISO-8859-1" >
<?xml version="1.0" standalone="no"? styles="poems.css">
```

[24] VersionInfo ::= S 'version' Eq (' VersionNum ' | " VersionNum ")

يقوم Production [24] بتعريف سلسلة معلومات الإصدار كمسافة بيضاء متبوعة بـإصدار سلسلة حرفية متبوع بعلامة المساواة (Production [25]) متبوع برقم إصدار محـاط بعلامـة اقتباس فردية أو مزدوجة. وفيما يلي سلاسل معلومات الإصدار الصحيحة:

```
version="1.0"
version='1.0'
version = '1.0'
```

وفيما يلي سلاسل معلومات الإصدار غير النحوية:

```
version='1.0"
"1.0"=version
```

[25] Eq ::= S? '=' S?

يقوم Production [25] بتعريف سلسلة Eq في التدقيق النحوي كرمز لعلامة المسـاواة (=) في المستندات. وقد تظهر المسافة البيضاء (Production [3]) وقد لا تظهر على أي جانب مـن علامة المساواة.

[26] VersionNum ::= ([a-zA-Z0-9_.:] | '-')+

يوضح Production [26] أن رقم الإصدار يتكون مـن حـرف أو أكـثر مـن a إلـى z والأحرف اللاتيني الكبيرة والتسطير أسفل السطر والنقطة والواصلة. يعتبر ما يلي أرقام إصـدار صحيحة نحوياً:

```
1.0
1.x
1.1.3
1.5EA2
v1.5
EA_B
```

يعتبر ما يلي أرقام إصدار غير نحوية:

```
version 1.5
```

1,5
1!1
1 5 3
v 1.5

_



يعتبر رقم الإصدار الوحيد المستخدم حالياً في مستندات XML هو 1.0. قد يتم قراءة هذا المنتج كما يلي أيضاً:

VersionNum ::= "1.0"

[27] Misc ::= Comment | PI | S

يقوم [27] Production بتعريف العناصر المتنوعة في مستند XML بما في ذلك التعليقات ([15] Production) وتعليمات التشغيل ([16] Production) والمسافة البيضاء ([3] Production).

تعريف نوع المستند

[28] doctypedecl ::= '<!DOCTYPE' S Name (S ExternalID)? S? ('[' (markupdecl | PEReference | S)* ']' S?)? '>'

يوضح [28] Production أن إعلان نوع المستند يتكون من السلسلة الحرفية <!DOCTYPE المتبوعة بمسافة بيضاء ([3] Production) متبوعة باسم XML ([5] Production) واختيارياً متبوعة بمسافة بيضاء وبرقم معرف خارجي ([75] Production) وبأكثر من مسافة متبوعة بقوس مربع مفتوح من ناحية اليسار ([) وبصفر أو بأكثر من إعلان لغة ترميز (Production [29]) ومراجع كينونة معامل ([69] Production) وبمسافة بيضاء متبوعة بقوس مربع مفتوح ناحية اليمين (]) وبمسافة بيضاء متبوعة بقوس زاوية إغلاق.

[29] markupdecl ::= elementdecl | AttlistDecl | EntityDecl | NotationDecl | PI | Comment

يوضح [29] Production أن إعلان لغة الترميز قد يكون إعلان عنصر (Production [45]) أو إعلان قائمة سمة ([52] Production) أو إعلان كينونة ([70] Production) أو إعلان رمز ([82] Production) أو تعليمات تشغيل ([16] Production) أو تعليق ([15] Production).

المجموعة الجزئية الخارجية

[30] extSubset ::= TextDecl? ExtSubsetDecl

يوضح Production [30] أن المجموعة الجزئية الخارجية تتكون من إعلان نـــص اختيـــاري (Production [77]) متبوع بإعلان مجموعة جزئية خارجية (Production [31]). لاحظ أنـــه يتم دمج المجموعات الجزئية الخارجية في المستند من الملفات الموجودة فيه قبل اختيـــار بنـــاء الجملة في مقابل التدقيق النحوي BNF.

[31] extSubsetDecl ::= (markupdecl | conditionalSect | PEReference | S)*

يوضح Production [31] أن إعلان المجموعة الجزئية الخارجية يحتوي على أي عدد مـــن إعلانات لغة الترميز (Production [29]) والمقاطع الشرطية (Production [61]) ومراجـــع كينونة المعامل (Production [69]) والمسافة البيضاء في أي ترتيـــب. تحتـــوي المجموعـــة الجزئية أساساً على أي شيء يمكن أن يحتوي عليه DTD الداخلي.

إعلان المستند المستقل بذاته

[32] SDDecl ::= S ئىstandalone' Eq ((""" ('yes' | 'no') """) | ("'" ('yes' | 'no') "'"))

يوضح Production [32] أن إعلان المستند المستقل بذاته يتكون من الحرف المســـتقل بذاتـــه متبوع بعلامة المساواة "التي قد تكون محاطة بمسافة بيضاء" متبوعة بإحدى القيمتين وهمـــا Yes أو NO داخل علامة اقتباس فردية أو مزدوجة. تحتوي إعلانات المستند المستقل بذاته الصحيحـــة على:

standalone="yes"
standalone="no"
standalone='yes'
standalone='no'
standalone="yes"
standalone="no"

تعريف اللغة

[33] LanguageID ::= Langcode ('-'Subcode)*

يقوم Production [33] بتعريف معرف اللغة كرمز لغة (Production [34]) متبوع بصفر أو بأكثر من واصلة ورموز فرعية (Production [38]).

[34] Langcode ::= ISO639Code | IanaCode | UserCode

يقوم Production [34] بتعريف رمز اللغة باعتباره رمز ISO 639 (Production [35]) ورمز IANA (Production [36]) أو رمز المستخدم (Production [37]).

[35] ISO639Code ::= ([a-z] | [A-Z]) ([a-z] | [A-Z])

يقوم Production [35] بتعريف رمز ISO 639 باعتباره حرفين صغيرين من الأبجدية الإنجليزية. يوجد (52 5 52) 2704 رمز ISO 639 النحوي بما في ذلك:

en
fr
jp
EN
jP
Fr

يوجد عدد لا نهائي من السلاسل التي لا تعتبر رموز ISO 639 النحوية بما في ذلك:

English
French
Japanese
""÷´êNn

[36] IanaCode ::= ('I' | 'I') '-' ([a-z] | [A-Z])+

يقوم Production [36] بتعريف رمز IANA كحرف I كبير أو صغير متبوع بواصلة متبوعة بحرف أو أكثر من الأبجدية الإنجليزية. يعتبر ما يلي رموز IANA النحوية:

i-no-bok
i-no-nyn
i-navajo
i-mingo

لا يعتبر ما يلي رموز IANA نحوية:

no-bok
no-nyn
navajo
mingo
i-""÷´êNn

[37] UserCode ::= ('x' | 'X') '-' ([a-z] | [A-Z])+

يقوم [37] Production بتعريف رمز المستخدم كحرف x صغير أو كبير متبوع بواصلة متبوعة بحرف أو أكثر من الأبجدية الإنجليزية. فيما يلي رموز المستخدم النحوية:

x-klingon
X-Elvish

لا يعتبر ما يلي رموز IANA نحوية:

Elvish
xklingon
´-´y¨êNn

[38] Subcode ::= ([a-z] | [A-Z])+

يقوم [38] Production بتعريف الرمز الفرعي كحرف أو أكثر كبير أو صغير من الأبجدية الإنجليزية. يعتبر ذلك رموز فرعية نحوية:

gb
GreatBritain
UK
Uk

لا يعتبر ذلك رموز فرعية نحوية:

Great Britain

العنصر

[39] element ::= EmptyElemTag | STag content Etag

يقوم [39] Production بتعريف العنصر باعتباره علامة عنصر فارغ ([44] production) أو علامة بدء ([40] production) متبوعة بمحتوى ([43] production) متبوعة بعلامة نهاية ([42] production).

يعتبر ما يلي عناصر صحيحة:

<P>Hello!</P>
<P/>
<P></P>

لا يعتبر ما يلي عناصر صحيحة:

<P>Hello!</p>
<P>
</Q>

علامة البدء:

[40] STag ::= '<' Name (S Attribute)* S? '>'

يوضح [40] Production أن علامة البدء تبدأ بالعلامة < متبوعة باسم XML (Production [5]) مفصولة بمسافة بيضاء متبوعة بإغلاق >. وفيما يلي بعض علامات البدء الصحيحة.

```
<DOCUMENT>
<ىتêîن÷EHT>
<DOCUMENT  >
<DOCUMENT TITLE="The Red Badge of Courage" >
<DOCUMENT TITLE="The Red Badge of Courage" PAGES="129">
```

وفيما يلي علامات البدء غير الصحيحة

```
< DOCUMENT>
< >
<12091998>
```

[41] Attribute ::= Name Eq AttValue

يوضح [41] Production أن السمة تتكون من اسم XML ([5] Production) متبوع بعلامة مساواة (قد تقع في مسافة بيضاء) متبوعة بقيمة سمة ([10] Production). تتضمن السمات النحوية ما يلي:

```
TITLE="The Red Badge of Courage"
PAGES="129"
TITLE = "The Red Badge of Courage"
PAGES = "129"
TITLE='The Red Badge of Courage'
PAGES='129'
SENTENCE='Jim said, "I didn&apos;t expect to see you here."'
```

بينما تتضمن السمات غير النحوية ما يلي:

```
TITLE="The Red Badge of Courage'
PAGES=129
SENTENCE='Then Jim said, "I didn't expect to see you here."'
```

علامة الانتهاء:

[42] ETag ::= '</' Name S? '>'

يقوم [42] Production بتعريف علامة الانتهاء كسلسلة حرفية /> متبوعة على الفور باسم XML متبوع اختيارياً بمسافة بيضاء متبوعة بحرف < وفيما يلي علامات انتهاء XML النحوية على سبيل المثال:

```
</PERSON>
</PERSON >
</AbrahamLincoln>
</ûîê÷ىEHT>
```

بينما يعتبر ما يلي علامات انتهاء XML غير النحوية:

```
</ PERSON>
</Abraham Lincoln>
</PERSON NAME="Abraham Lincoln">
</>
```

محتوى العناصر:

[43] content ::= (element | CharData | Reference | CDSect | PI | Comment)*

يقوم [43] Production بتعريف المحتوى كأي عـــدد مـــن العنـــاصر ([39] Production) وبيانات الأحرف ([14] Production) والمراجع CDATA ,([67] Production) وتعليمـــات التشـــغيل ([18] Production) وتعليمـــات التشـــغيل ([16] Production) والتعليقـــات ([15] Production) بأي ترتيب. يسرد هذا الإنتاج كل ما قد يظهر في العنصر.

علامات العناصر الفارغة

[44] EmptyElemTag ::= '<' Name (S Attribute)* S? '/>'

يقوم ([44] Production) بتعريف علامة العنصر الفارغة كحرف > متبوع باسم XML متبـوع بمسافة بيضاء متبوعة بسمات صفر مفصولة عن بعضها بمسافة بيضاء المتبوعة بمسافة بيضـــاء اختيارياً متبوعة بحرف </ وفيما يلي بعض العلامات النحوية الفارغة:

```
<PERSON/>
<PERSON />
<Person/>
```

<person />
<AbrahamLincoln/>
<person />

وفيما يلي العلامات الفارغة غير النحوية:

< PERSON/>
<PERSON>
</Person>
</person/>
</>

"تعتبر كل من علامتي البدء والانتهاء الثانية والثالثة هما علامتان نحويتان".

إعلان نوع العنصر:

[45] elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'

يوضح [45] Production أن إعلان العنصر يتكون من الحرف <!ELEMENT متبوع بمسافة بيضاء ومتبوع باسم XML ([5] Production) المتبوع اختيارياً بمسافة بيضاء متبوعة بحرف >.

تتضمن إعلانات العنصر النحوية ما يلي:

<!ELEMENT DOCUMENT ANY>
<!ELEMENT HR EMPTY>
<!ELEMENT DOCUMENT (#PCDATA | P | H)>

[46] contentspec ::= 'EMPTY' | 'ANY' | Mixed | children

يقوم [46] Production بتعريف تعيين محتوى كالحرفين EMPTY أو ANY وكقائمة توابع ([47] Production) أو كمحتوى مختلط ([51] Production).

أنواع محتوى العنصر

[47] children ::= (choice | seq) ('?' | '*' | '+')?

يوضح [47] Production أن قائمة التوابع تتكون من اختيار ([49] Production) أو تسلسل ([50] Production) المتبوع اختيارياً بأحد الحرف التالية ? أو * أو +.

[48] cp ::= (Name | choice | seq) ('?' | '*' | '+')?

يقوم [48] Production بتعريف جزء المحتوى كاسم XML ([5] Production) أو اختيار ([49] Production) أو تسلسل ([50] Production) الذي قد يلحقه ? أو * أو +

[49] choice ::= '(' S? cp (S? '|' S? cp)* S? ')'

يوضح Production [49] أن الاختيار هو أحد أجزاء المحتــوى أو أكــثر (Production [48]) داخل أقواس ومفصول عن بعضه بأشرطة عمودية ومسافة بيضـــاء اختياريــة تتضمــن الاختيارات النحوية ما يلي:

(P | UL | H1 | H2 | H3 | H4 | H5 | BLOCKQUOTE | PRE | HR | DIV)
(P|UL|H1|H2|H3|H4|H5|H6|BLOCKQUOTE|PRE|HR|DIV)
(SON | DAUGHTER)
(SON | DAUGHTER)
(ADDRESS | (NAME, STREET, APT, CITY, STATE, ZIP))

[50] seq ::= '(' S? cp (S? ',' S? cp)* S? ')'

يوضح Production [50] أن التسلسل هو واحد أو أكثر من أجزاء المحتوى (Production [48]) محاط في أقواس ومفصول عن بعضه بواسطة فواصل ومسافات بيضاء اختيارية تتضمـن التسلسلات النحوية ما يلي:

(NAME, STREET, APT, CITY, STATE, ZIP)
(NAME , STREET , APT , CITY , STATE , ZIP)
(NAME,STREET,APT,CITY,STATE,ZIP)
(NAME,STREET,APT, CITY,STATE,ZIP)
(NAME, (STREET|BOX), (APT|SUITE), CITY, STATE, ZIP, COUNTRY?)
(NAME)

إعلان المحتوى المختلط

[51] Mixed ::= '(' S? '#PCDATA' (S? '|' S? Name)* S? ')*' | '(' S? '#PCDATA' S? ')'

يوضح Production [51] إن المحتوى المختلط هو الحرف (#PCDATA) (الذي يحتــوي على تصريحات للمسافة البيضاء الاختيارية) أو على اختيار يحتوي علـــى حـــرف #PCDATA كجزء المحتوى الأول. وفيما يلي بعض أنواع المحتوى المختلط النحوية:

(#PCDATA)
(#PCDATA)
(#PCDATA | PERSON)
(#PCDATA | PERSON)
(#PCDATA | TITLE | JOURNAL | MONTH | YEAR | SERIES | VOLUME)

وفيما يلي أنواع المحتوى المختلط غير النحوية:

```
(PERSON | #PCDATA)
(#PCDATA, TITLE, #PCDATA, JOURNAL, MONTH, YEAR, #PCDATA)
(#PCDATA | (NAME, STREET, APT, CITY, STATE, ZIP))
```

إعلان قائمة السمة:

[52] AttlistDecl ::= '<!ATTLIST' S Name AttDef* S? '>'

يوضح [52] Production أن إعلان قائمة السمة يتكون من الحرف <!ATTLIST متبوع بمسافة بيضاء متبوعة باسم XML ([5] Production) متبوع بصفر أو بتعريفات سمات أكثر ([53] Production) متبوعة بمسافة بيضاء متبوعة بحرف >.

تتضمن إعلانات قائمة السمات النحوية ما يلي:

```
<!ATTLIST IMG ALT CDATA #REQUIRED>
<!ATTLIST AUTHOR EXTENSION CDATA #IMPLIED>
<!ATTLIST AUTHOR COMPANY   CDATA #FIXED "TIC">
<!ATTLIST P VISIBLE (TRUE | FALSE) "TRUE">
<!ATTLIST ADDRESS STATE NMTOKEN #REQUIRED>
<!ATTLIST ADDRESS STATES NMTOKENS #REQUIRED>
<!ATTLIST P PNUMBER ID #REQUIRED>
<!ATTLIST PERSON FATHER IDREF #IMPLIED>
<!ATTLIST SLIDESHOW SOURCES ENTITIES #REQUIRED>
<!ATTLIST SOUND PLAYER NOTATION (MP) #REQUIRED>
```

[53] AttDef ::= S Name S AttType S DefaultDecl

يقوم [53] Production بتحديد تعريف سمة كمسافة بيضاء واسم XML (Production [5]) ومسافة بيضاء أكبر ونوع سمة ([54] Production) ومسافة بيضاء أكبر وإعلان افتراضي ([60] Production). تتضمن تعريفات السمات النحوية ما يلي:

```
IMG ALT CDATA #REQUIRED
AUTHOR EXTENSION CDATA #IMPLIED
AUTHOR COMPANY   CDATA #FIXED "TIC"
P VISIBLE (TRUE | FALSE) "TRUE"
ADDRESS STATE NMTOKEN #REQUIRED
ADDRESS STATES NMTOKENS #REQUIRED
```

```
P PNUMBER ID #REQUIRED
PERSON FATHER IDREF #IMPLIED
SLIDESHOW SOURCES ENTITIES #REQUIRED
SOUND PLAYER NOTATION (MP) #REQUIRED
```

أنواع السمات

[54] AttType ::= StringType | TokenizedType | EnumeratedType

يقوم [54] Production بتعريف نوع السمة كنوع سلسلة ([55] Production) أو نوع متعدد ([57] Production).

[55] StringType ::= 'CDATA'

يقوم نوع سلسلة كالحرف CDATA.

[56] TokenizedType ::= 'ID' | 'IDREF' | 'IDREFS' | 'ENTITY' | 'ENTITIES' | 'NMTOKEN' | 'NMTOKENS'

يقوم [56] Production بتعريف TokenizedType كأي من هذه الحرف السبعة:

ID
IDREF
IDREFS
ENTITY
ENTITIES
NMTOKEN
NMTOKENS

أنواع السمات المتعددة

[57] EnumeratedType ::= NotationType | Enumeration

يقوم [57] Production بتعريف النوع المتعدد كنوع الرمز ([58] Production) أو التعدد ([59] Production).

[58] NotationType ::= 'NOTATION' S '(' S? Name (S? '|' S? Name)* S? ')'

يقوم ([58] Production) بتعريف نوع الرمز كحرف NOTATION متبوع بمسافة بيضاء متبوعة باسم XML واحد أو أكثر ([5] Production) مفصول بأشرطة عمودية ومحاط بأقواس.
فيما يلي بعض أنواع الرموز النحوية:

NOTATION (MP)
NOTATION (MP | PDF)
NOTATION (mp | gcc | xv)
NOTATION (A | B | C)

وفيما يلي بعض أنواع الرموز النحوية:

NOTATION ("MP")
NOTATION (MP PDF)
NOTATION (mp, gcc, xv)
NOTATION ("A" "B" "C")

[59] Enumeration ::= '(' S? Nmtoken (S? '|' S? Nmtoken)* S? ')'

يقوم (Production [59]) بتعريف تعدد كأحد رموز أسماء XML أو أكثر (Production [7]) مفصول بأشرطة عمودية ومحاط بأقواس فيما يلي بعض التعددات النحوية:

(airplane)
(airplane | train | car | horse)
(airplane | train | car | horse)
(cavalo | carro | trem |avi‹o)
(A | B | C | D | E | F | G | H)

لا يعتبر ما يلي تعدادي مصرح بها:

()
(airplane train car horse)
(A, B, C, D, E, F, G, H)
airplane | train | car | horse

افتراضات السمات

[60] DefaultDecl ::= '#REQUIRED' | '#IMPLIED' | (('#FIXED' S)? AttValue)

يقوم [60] Production بتعريف الإعلان الافتراضي كأحد الأشياء الأربعة التالية:

- الحرف #REQUIRED.
- الحرف #IMPLIED.
- الحرف #FIXED متبوع بمسافة بيضاء (Production [3]) متبوعة بقيمة سمة (Production [10]).

♦ قيمة سمة ([10] Production).

المقطع الشرطي

[61] conditionalSect ::= includeSect | ignoreSect

يقوم [61] Production بتعريف المقطع الشرطي كمقطـــع تضميـــن (62] Production) أو كمقطع تجاهل (ignore section) ([63] Production).

[62] includeSect ::= '<![' S? 'INCLUDE' S? '[' extSubsetDecl ']]>'

يقـــوم [62] Production بتعريـــف مقطـــع تضميـــن ([31] Production) موجـــودة <![INCLUDE[ ]]>، modulo whitespace.

وفيما يلي مقاطع التضمين ([31] Production) النحوية:

<![INCLUDE [ ]]>

<![INCLUDE[ ]]>

<![INCLUDE[ ]]>

[63] ignoreSect ::= '<![' S? 'IGNORE' S? '[' ignoreSectContents* ']]>'

يقوم [63] Production بتعريف مقطع تجـــاهل (ignore) كمحتويـــات مقطـــع التجـــاهل ([64] Production) الموجودة بين <![IGNORE[ ]]>, modulo whitespace. وفيما يلي مقاطع التجاهل (ignore) النحوية:

<![IGNORE [ ]]>

<![IGNORE[ ]]>

<![IGNORE[ ]]>

[64] ignoreSectContents ::= Ignore ('<![' ignoreSectContents ']]>' Ignore)*

يقوم ([64] Production) بتعريف محتويات مقاطع التجاهل ككتلة تجـــاهل (Production [65]) المتبوعة بنص اختياري موجود بين أحرف <![and]]> المتبوع بنص أكبر، قـــد يتـــم تكرار ذلك حسب رغبتك.

[65] Ignore ::= Char* - (Char* ('<![' | ']]>') Char*)

يقوم (Production 65) بتعريف كتلة تجاهل (ignore block) كأي تشغيل للنـــص الـــذي يحتوي على أي من أحرف <![or]]> مما يؤدي إلى استحالة حدوث أي اختلاف من مكـــان انتهاء كتلة تجاهل.

مرجع الحرف

[66] CharRef ::= '&#[0-9]+ ';' | '&#x[0-9a-fA-F]+ ';'

يقوم Production [66] بتعريف نموذجين لمراجع الأحرف يعتبر النموذج الأول هو حرف &# متبوع برقم ASCII أو أكثر من ٥ إلى ٩ أما النموذج الثاني فهو الحـــرف &#x متبـــوع برقـــم عشري سداسي أو أكثر من ٥ إلى F. قد تكون الأرقام من ١٥ إلى ١٦ وهي (الأحرف مـــن A إلى F) أحرف لاتيني صغيرة أو كبيرة.

مرجع الكينونة

[67] Reference ::= EntityRef | CharRef

يقوم Production [67] بتعريف المرجع كمرجع كينونـــة ([68] Production) أو كمرجـــع حرف ([66] Production).

[68] EntityRef ::= '&' Name ';'

يقوم Production [68] بتعريف مرجع الكينونة كاسم XML ([5] Production) موجـــود بين حرف & وفاصلة منقوطة. وفيما يلي مراجع الكينونة النحوية:

```
&amp;
&agrave;
&my_abbreviation;
```

وفيما يلي مراجع الكينونة غير النحوية:

```
&amp
& agrave ;
& my_abbreviation;
```

[69] PEReference ::= '%' Name ';'

يقوم Production [69] بتعريف مرجع كينونة المعامل كاســـم XML ([5] Production) الموجود بين حرف النسبة المئوية والفاصلة المنقوطة. وفيما يلي مراجع كينونة المعامل النحوي:

```
%inlines;
%mathml;
%MyElements;
```

وفيما يلي مراجع كينونة المعامل غير النحوية:

```
%inlines
% mathml ;
%my elements;
```

إعلان الكينونة

[70] EntityDecl ::= GEDecl | PEDecl

يقوم [70] Production بتعريف إعلان كينونة كإعلان كينونة عـــام ([71] Production) أو كإعلان كينونة معامل ([71] Production).

[71] GEDecl ::= '<!ENTITY' S Name S EntityDef S? '>'

يقوم [71] Production بتعريف إعلان كينونة عام كحرف <!ENTITY متبـــوع بمســـافة بيضاء ([3] Production) متبوعة باسم XML ([5] Production) متبـــوع بتعريـــف كينونـــة [73] Production المتبوع بمسافة بيضاء اختيارية متبوعة بحرف < وفيما يلي بعض إعلانات الكينونة العامة النحوية:

```
<!ENTITY alpha #&"945;">
<!ENTITY Alpha #&"913;">
<!ENTITY SPACEMUSIC SYSTEM "/sounds/space.wav" NDATA MP >
<!ENTITY LOGO SYSTEM "logo.gif"<>
<!ENTITY COPY99 "Copyright 1999 %erh;">
```

وفيما يلي بعض إعلانات الكينونة العامة غير النحوية:

```
<!ENTITY alpha &#945;>
<!ENTITY Capital Greek Alpha #&"913;">
<!ENTITY LOGO SYSTEM logo.gif>
```

[72] PEDecl ::= '<!ENTITY' S '%' S Name S PEDef S? '>'

يقوم [72] Production بتعريف إعلان كينونة معامل كحرف <!ENTITY متبوع بمســـافة بيضاء ([3] Production) متبوعة بعلامة نسبة مئوية ومسافة بيضاء أكبر متبوعة باســـم XML ([5] Production) متبوع بتعريف كينونـــة ([73] Production) متبوعـــة بمســـافة بيضـــاء اختيارية متبوعة بحرف < مما يوضح أن إعلانات كينونة المعامل هي نفس إعلانات الكينونـــة العامة فيما عدا % بين <!ENTITY والاسم. وفيما يلي بعض إعلانات كينونة المعامل النحوية:

```
<!ENTITY % fulldtd "IGNORE">
<!ENTITY % ERH "Elliotte Rusty Harold">
<!ENTITY % inlines
  "(person | degree | model | product | animal | ingredient)*">
```

وفيما يلي بعض إعلانات كينونة المعامل غير النحوية:

```
<!ENTITY %fulldtd; "IGNORE">
<!ENTITY % ERH  Elliotte Rusty Harold>
<!ENTITY % inlines
 "(person | degree | model | product | animal | ingredient)*'>
```

[73] EntityDef ::= EntityValue | (ExternalID NDataDecl?)

يوضح [73] Production أن تعريف الكينونة هو قيمة الكينونة ([9] Production) أو ID خارجي ([75] Production) متبوع بإعلان ([76] Production).

[74] PEDef ::= EntityValue | ExternalID

يوضح [74] Production إن تعريف كينونة المعامل قد يكون قيمة كينونة (Production [9]) أو ID خارجي ([75] Production).

إعلان كينونة خارجي

[75] ExternalID ::= 'SYSTEM' S SystemLiteral | 'PUBLIC' S PubidLiteral S SystemLiteral

يقوم [75] Production بتعريف ID خارجي كالكلمة الأساسية SYSTEM متبوعة بمسافة بيضاء وحرف نظام ([11] Production) أو الكلمة الأساسية PUBLIC متبوعة بمسافة بيضاء وحرف ID عام ([12] Production) ومسافة بيضاء أكبر وحرف نظام ([11] Production). وفيما يلي بعض IDs الخارجية النحوية:

```
SYSTEM "logo.gif"
SYSTEM "/images/logo.gif"
SYSTEM "http://www.idgbooks.com/logo.gif"
SYSTEM "../images/logo.gif"
PUBLIC "-//IETF//NONSGML Media Type image/gif//EN"
    "http://www.isi.edu/in-notes/iana/assignments/media-
types/image/gif"
```

وفيما يلي بعض IDs الخارجية غير النحوية:

```
SYSTEM logo.gif
SYSTEM "/images/logo.gif'
SYSTEM http://www.idgbooks.com/logo.gif
PUBLIC "-//IETF//NONSGML Media Type image/gif//EN"
```

```
PUBLIC "http://www.isi.edu/in-notes/iana/assignments/media-types/image/gif"
[76] NDataDecl  ::= S 'NDATA' S Name
```

يقوم [76] Production بتعريف إعــلان NData كمســافة بيضـــاء ([3] Production) متبوعة بحرف NDATA متبوع بمسافة بيضاء متبوعة باسم XML ([5] Production).

```
NDATA PDF
NDATA MIDI
```

إعلان النص

[77] TextDecl ::= '<?xml' VersionInfo? EncodingDecl S? '?>'

يوضح ([77] Production) أن إعلان النـــص يبــدو كــإعلان XML ([23] Production) باستثناء أنه قد لا يحتوي على إعلان مستند مستقل بذاتـــه ([32] Production). وفيمـــا يلـــي إعلانات النص النحوية:

```
<?xml version="1.0"?>
<?xml version="1.0" encoding="Big5"?>
<?xml version="1.0" encoding="ISO-8859-5"?>
```

لا يعتبر ما يلي إعلانات نص نحوية:

```
<?xml?>
<?xml encoding="Big5"?>
<?xml encoding="Big5" version="1.0" ?>
<?xml version="1.0" standalone="yes"? encoding="ISO-8859-1" >
<?xml version="1.0" styles="poems.css">
<?xml version="1.0" encoding=":ISO-8859-1" standalone="yes"?>
<?xml version="1.0" standalone="no"? >
```

كينونة موزعة خارجية جيدة الصياغة:

[78] extParsedEnt ::= TextDecl? content

يوضح ([78] Production) أن الكينونة الموزعة العامة الخارجية تتكون من إعلان نـــص اختياري متبوع بمحتوى ([43] Production) يعتبر أساس هذا الإنتاج هو أن المحتوى قـــد لا يتضمن DTD أو أي إعلانات لغة ترميز.

[79] extPE ::= TextDecl? extSubsetDecl

يوضح ([79] Production) أن كينونة المعامل الخارجية تتكون من إعلان نص اختيـــاري متبوع بإعلان مجموعة جزئية خارجية ([31] Production).

إعلان تشفير: Encoding Declaration

[80] EncodingDecl ::= S 'encoding' Eq ('"' EncName '"' | "'" EncName "'")

يقوم ([80] Production) بتعريف إعلان التشفير كمسافة بيضاء ([3] Production) متبوعـــة بسلسلة "encoding" متبوعة بعلامة مساواة ([25] Production) محاط بعلامة اقتباس فردية أو مزدوجة. يعتبر ما يلي إعلانات تشفير صحيحة:

```
encoding="Big5"
encoding="ISO-8859-5"
encoding = "Big5"
encoding = "ISO-8859-5"
encoding= 'Big5'
encoding= 'ISO-8859-5'
```

وفيما يلي إعلانات التشفير غير الصحيحة:

```
encoding "Big5"
encoding="ISO-8859-51'
encoding = "Big5
encoding = 'ISO-8859-5"
```

[81] EncName ::= [A-Za-z] ([A-Za-z0-9._] | '-')*

يوضح ([81] Production) أن اسم التشفير يبدأ بأحد أحرف ASCII من A إلى Z أو من a إلى z متبوعة بأي عدد من أحرف وأرقام ونقاط وأسطر وواصلات ASCII. يعتبر ما يلي أسماء التشفير الصحيحة:

```
ISO-8859-1
Big5
GB2312
```

وفيما يلي أسماء التشفير غير النحوية:

```
ISO 8859-1
Big5 Chinese
```

GB 2312
Eλor851

إعلانات الرمز

[82] NotationDecl ::= '<!NOTATION' S Name S (ExternalID | PublicID) S? '>'

يقوم ([82] Production) بتعريف إعلان رمز كالسلسلة الحرفية <!NOTATION المتبوعة بمسافة بيضاء ([3] Production) المتبوعة باسم XML ([5]Production) للرمز المتبوع بمسافة بيضاء متبوعة برقم معرف خارجي ([75] Production) أو ID عام (Production [83]) متبوع بمسافة بيضاء اختيارية متبوعة بسلسلة حرفية ">". وفيما يلي إعلانات رموز نحوية:

```
<!NOTATION GIF SYSTEM "image/gif">
<!NOTATION GIF SYSTEM "image/gif" >
<!NOTATION GIF PUBLIC
  "-//IETF//NONSGML Media Type image/gif//EN"
  "http://www.isi.edu/in-notes/iana/assignments/media-
types/image/gif">
```

وفيما يلي إعلانات الرموز غير النحوية:

```
<! NOTATION GIF SYSTEM "image/gif" >
< !NOTATION GIF SYSTEM "image/gif" >
<!NOTATION GIF "image/gif">
<!NOTATION GIF SYSTEM image/gif>
<!NOTATION GIF PUBLIC
  "http://www.isi.edu/in-notes/iana/assignments/media-
types/image/gif">
```

[83] PublicID ::= 'PUBLIC' S PubidLiteral

يقوم [83] Production بتعريف ID عام كسلسلة حرفية متبوعة بمسافة بيضاء ([3] Production) متبوعة بحرف ID العام ([12] Production). وفيما يلي أرقام المعرفات العامة النحوية: (IDs)

```
PUBLIC "-//IETF//NONSGML Media Type image/gif//EN"
```

PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN//XML"

وفيما يلي IDs العامة غير النحوية

PUBLIC -//IETF//NONSGML Media Type image/gif//EN
PUBLIC 'ISO 8879:1986//ENTITIES Added Latin 1//EN//XML"

الأحرف

[84] Letter ::= BaseChar | Ideographic

يقوم [84] Production بتعريف الحرف كحرف قاعدة أو حرف ideographic:

[85] BaseChar ::= [#x0041-#x005A] | [#x0061-#x007A] | [#x00C0-#x00D6] | [#x00D8-#x00F6] | [#x00F8-#x00FF] | [#x0100-#x0131] | [#x0134-#x013E] | [#x0141-#x0148] | [#x014A-#x017E] | [#x0180-#x01C3] | [#x01CD-#x01F0] | [#x01F4-#x01F5] | [#x01FA-#x0217] | [#x0250-#x02A8] | [#x02BB-#x02C1] | #x0386 | [#x0388-#x038A] | #x038C | [#x038E-#x03A1] | [#x03A3-#x03CE] | [#x03D0-#x03D6] | #x03DA | #x03DC | #x03DE | #x03E0 | [#x03E2-#x03F3] | [#x0401-#x040C] | [#x040E-#x044F] | [#x0451-#x045C] | [#x045E-#x0481] | [#x0490-#x04C4] | [#x04C7-#x04C8] | [#x04CB-#x04CC] | [#x04D0-#x04EB] | [#x04EE-#x04F5] | [#x04F8-#x04F9] | [#x0531-#x0556] | #x0559 | [#x0561-#x0586] | [#x05D0-#x05EA] | [#x05F0-#x05F2] | [#x0621-#x063A] | [#x0641-#x064A] | [#x0671-#x06B7] | [#x06BA-#x06BE] | [#x06C0-#x06CE] | [#x06D0-#x06D3] | #x06D5 | [#x06E5-#x06E6] | [#x0905-#x0939] | #x093D | [#x0958-#x0961] | [#x0985-#x098C] | [#x098F-#x0990] | [#x0993-#x09A8] | [#x09AA-#x09B0] | #x09B2 | [#x09B6-#x09B9] | [#x09DC-#x09DD] | [#x09DF-#x09E1] | [#x09F0-#x09F1] | [#x0A05-#x0A0A] | [#x0A0F-#x0A10] | [#x0A13-#x0A28] | [#x0A2A-#x0A30] | [#x0A32-#x0A33] | [#x0A35-#x0A36] | [#x0A38-#x0A39] | [#x0A59-#x0A5C] | #x0A5E | [#x0A72-#x0A74] | [#x0A85-#x0A8B] | #x0A8D | [#x0A8F-#x0A91] | [#x0A93-#x0AA8] | [#x0AAA-#x0AB0] | [#x0AB2-#x0AB3] | [#x0AB5-#x0AB9] | #x0ABD | #x0AE0 | [#x0B05-#x0B0C] | [#x0B0F-#x0B10] | [#x0B13-#x0B28] | [#x0B2A-#x0B30] | [#x0B32-#x0B33] | [#x0B36-#x0B39] | #x0B3D | [#x0B5C-#x0B5D] | [#x0B5F-#x0B61] | [#x0B85-#x0B8A] | [#x0B8E-#x0B90] | [#x0B92-#x0B95] | [#x0B99-#x0B9A] | #x0B9C | [#x0B9E-#x0B9F] | [#x0BA3-#x0BA4] | [#x0BA8-#x0BAA] | [#x0BAE-#x0BB5] | [#x0BB7-#x0BB9] | [#x0C05-#x0C0C] | [#x0C0E-#x0C10] | [#x0C12-#x0C28] | [#x0C2A-#x0C33] | [#x0C35-#x0C39] | [#x0C60-#x0C61] | [#x0C85-#x0C8C] | [#x0C8E-#x0C90] | [#x0C92-#x0CA8] | [#x0CAA-#x0CB3] | [#x0CB5-#x0CB9] | #x0CDE | [#x0CE0-#x0CE1] | [#x0D05-

#x0D0C] | [#x0D0E-#x0D10] | [#x0D12-#x0D28] | [#x0D2A-#x0D39] | [#x0D60-#x0D61] | [#x0E01-#x0E2E] | #x0E30 | [#x0E32-#x0E33] | [#x0E40-#x0E45] | [#x0E81-#x0E82] | #x0E84 | [#x0E87-#x0E88] | #x0E8A | #x0E8D | [#x0E94-#x0E97] | [#x0E99-#x0E9F] | [#x0EA1-#x0EA3] | #x0EA5 | #x0EA7 | [#x0EAA-#x0EAB] | [#x0EAD-#x0EAE] | #x0EB0 | [#x0EB2-#x0EB3] | #x0EBD | [#x0EC0-#x0EC4] | [#x0F40-#x0F47] | [#x0F49-#x0F69] | [#x10A0-#x10C5] | [#x10D0-#x10F6] | #x1100 | [#x1102-#x1103] | [#x1105-#x1107] | #x1109 | [#x110B-#x110C] | [#x110E-#x1112] | #x113C | #x113E | #x1140 | #x114C | #x114E | #x1150 | [#x1154-#x1155] | #x1159 | [#x115F-#x1161] | #x1163 | #x1165 | #x1167 | #x1169 | [#x116D-#x116E] | [#x1172-#x1173] | #x1175 | #x119E | #x11A8 | #x11AB | [#x11AE-#x11AF] | [#x11B7-#x11B8] | #x11BA | [#x11BC-#x11C2] | #x11EB | #x11F0 | #x11F9 | [#x1E00-#x1E9B] | [#x1EA0-#x1EF9] | [#x1F00-#x1F15] | [#x1F18-#x1F1D] | [#x1F20-#x1F45] | [#x1F48-#x1F4D] | [#x1F50-#x1F57] | #x1F59 | #x1F5B | #x1F5D | [#x1F5F-#x1F7D] | [#x1F80-#x1FB4] | [#x1FB6-#x1FBC] | #x1FBE | [#x1FC2-#x1FC4] | [#x1FC6-#x1FCC] | [#x1FD0-#x1FD3] | [#x1FD6-#x1FDB] | [#x1FE0-#x1FEC] | [#x1FF2-#x1FF4] | [#x1FF6-#x1FFC] | #x2126 | [#x212A-#x212B] | #x212E | [#x2180-#x2182] | [#x3041-#x3094] | [#x30A1-#x30FA] | [#x3105-#x312C] | [#xAC00-#xD7A3]

يقوم [85] Production بسرد الأحرف الأساسية (Unicode characters) ويعتبر ما سبق هو الأحرف المعرفة Unicode والأبجدية ولكنها ليست علامات ترقيم أو أرقـــام علـــى ســـبيل المثال، يعتبر كل من A-Z وa-z هي أحرف أساسية وأن كل مـــن 0-9 و! و" و# و$ وهكـــذا ليست كذلك. تبدو هذه القائمة طويلة جداً نظراً لأنها تحتـــوي علـــى أحـــرف ليســـت الأبجديـــة الإنجليزية فقط ولكن تحتـــوي أيضـــاً علـــى الأبجديـــات التاليـــة Greek وHebrew وarabic وCyrillic ولكل أبجديات لغات Unicode الأخرى.

[86] Ideographic ::= [#x4E00-#x9FA5] | #x3007 | [#x3021-#x3029]

يقوم [86] Production بسرد أحرف ideographic يعتبر #x4E00-#x9FA5 هي:

Unicode's Chinese-Japanese-Korean unified ideographs. #x3007 يعتـــبر ideographic number zero تعتبر الأحرف من #x3021 إلى #x3029 هي أرقام الأسلوب Hangzhou.

[87] CombiningChar ::= [#x0300-#x0345] | [#x0360-#x0361] | [#x0483-#x0486] | [#x0591-#x05A1] | [#x05A3-#x05B9] | [#x05BB-#x05BD] | #x05BF | [#x05C1-#x05C2] | #x05C4 | [#x064B-#x0652] | #x0670 | [#x06D6-#x06DC] | [#x06DD-#x06DF] | [#x06E0-#x06E4] |

[#x06E7-#x06E8] | [#x06EA-#x06ED] | [#x0901-#x0903] | #x093C | [#x093E-#x094C] | #x094D | [#x0951-#x0954] | [#x0962-#x0963] | [#x0981-#x0983] | #x09BC | #x09BE | #x09BF | [#x09C0-#x09C4] | [#x09C7-#x09C8] | [#x09CB-#x09CD] | #x09D7 | [#x09E2-#x09E3] | #x0A02 | #x0A3C | #x0A3E | #x0A3F | [#x0A40-#x0A42] | [#x0A47-#x0A48] | [#x0A4B-#x0A4D] | [#x0A70-#x0A71] | [#x0A81-#x0A83] | #x0ABC | [#x0ABE-#x0AC5] | [#x0AC7-#x0AC9] | [#x0ACB-#x0ACD] | [#x0B01-#x0B03] | #x0B3C | [#x0B3E-#x0B43] | [#x0B47-#x0B48] | [#x0B4B-#x0B4D] | [#x0B56-#x0B57] | [#x0B82-#x0B83] | [#x0BBE-#x0BC2] | [#x0BC6-#x0BC8] | [#x0BCA-#x0BCD] | #x0BD7 | [#x0C01-#x0C03] | [#x0C3E-#x0C44] | [#x0C46-#x0C48] | [#x0C4A-#x0C4D] | [#x0C55-#x0C56] | [#x0C82-#x0C83] | [#x0CBE-#x0CC4] | [#x0CC6-#x0CC8] | [#x0CCA-#x0CCD] | [#x0CD5-#x0CD6] | [#x0D02-#x0D03] | [#x0D3E-#x0D43] | [#x0D46-#x0D48] | [#x0D4A-#x0D4D] | #x0D57 | #x0E31 | [#x0E34-#x0E3A] | [#x0E47-#x0E4E] | #x0EB1 | [#x0EB4-#x0EB9] | [#x0EBB-#x0EBC] | [#x0EC8-#x0ECD] | [#x0F18-#x0F19] | #x0F35 | #x0F37 | #x0F39 | #x0F3E | #x0F3F | [#x0F71-#x0F84] | [#x0F86-#x0F8B] | [#x0F90-#x0F95] | #x0F97 | [#x0F99-#x0FAD] | [#x0FB1-#x0FB7] | #x0FB9 | [#x20D0-#x20DC] | #x20E1 | [#x302A-#x302F] | #x3099 | #x309A

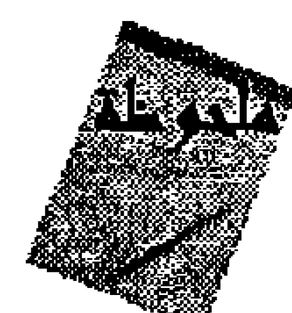
يقوم [87] Production بسرد أحرف الربط. تعتبر هذه الأحرف هي الأحـــرف المرتبطـة بالحرف السابق لتكوين شكل الحرف الفردي. على سبيل المثال، يعتبر حرف &#x300 هو the combining accent grave قد يتم تمثيل الحرف (a) المتبوع بواسطة: combining accent grave باعتباره a ويشـــغل عــرض حــرف فــردي واحــد حتــى إذا كــان بخــط monospaced.

[88] Digit ::= [#x0030-#x0039] | [#x0660-#x0669] | [#x06F0-#x06F9] | [#x0966-#x096F] | [#x09E6-#x09EF] | [#x0A66-#x0A6F] | [#x0AE6-#x0AEF] | [#x0B66-#x0B6F] | [#x0BE7-#x0BEF] | [#x0C66-#x0C6F] | [#x0CE6-#x0CEF] | [#x0D66-#x0D6F] | [#x0E50-#x0E59] | [#x0ED0-#x0ED9] | [#x0F20-#x0F29]

يقوم [88] Production بسرد الأحرف التي تعتبر أرقام وهي لا تتضمن الأرقام الأوروبيـة المعتادة فقط وهي ٥ ١ و٢ و٣ و٤ و٥ و٦ و٧ و٨ و٩ ولكنها تتضمن أيضاً الأرقـــام العربيـة الهندية المستخدمة أساساً في اللغة العربية المصرية، كما يتم استخدام الأرقــام الهنديــة العربيـة الشرقية في Persian وUrdu وغيرها.

[89] Extender ::= #x00B7 | #x02D0 | #x02D1 | #x0387 | #x0640 | #x0E46 | #x0EC6 | #x3005 | [#x3031-#x3035] | [#x309D-#x309E] | [#x30FC-#x30FE]

يقوم Production [89] بسرد الأحرف التي تعتبر إمتدادات وهي بـــالترتيب middle dot و the modifier letter triangular colon و the modi-fier letter half-triangular و colon و the Greek middle dot و the Arabic tatweel و the Thai maiyamok و the و Lao ko la و the ideographic iteration mark و five Japanese Kana repeat و marks و the Japanese Hiragana iteration mark و voiced iteration mark و the Japanese Katakana و Hiragana sound mark و prolonged sound mark. لا يعتـــبر الامتداد حرف أو حرف ربط ولكنه موجود في الكلمات كجزء منها. تعتبر الواصلة هي أقـــرب مرادف في الإنجليزية يتم استخدامها في كلمات مثل mother-in-law or well-off. ومع ذلـك، لا تعتبر الواصلة امتداد في XML.



يتم إزالة #x0387 وهي triangular colon من فئة الامتداد في أخر Unicode errata sheet و لكن لا يتم إدراجها في XML.

قيود جيدة الصياغة

وفقاً لتعيين XML 1.0، يعتبر مستند XML جيد الصياغة إذا:

١- تطابق مع مستند تسمية الإنتاج.

٢- تتطابق مع القيود الجيدة الصياغة المتوفرة في هذا التعيين.

٣- إذا كانت كلاً من الكينونات الموزعة التي تم الإشارة إليها مباشرة أو بشكل غير مباشـــر مع المستند جيدة الصياغة.

يتم تصميم موضوع المرجع لفهم ثاني هذه المتطلبات كما يحدد ما إذا كانت المستندات تتطابق مع هذا المتطلب.

ما هو القيد المصاغ جيداً؟

وبعد قراءة التدقيق النحوي BNF للغة XML 1.0، يجب ملاحظة إرفاق بعض المنتجات بـــالقيود المصاغة بشكل جيد وWFC المختصرة على سبيل المثال، يعتبر production [40] هو ما يلي:

```
[40] STag ::= '<' Name (S Attribute)* S? '>'
            [ WFC: Unique Att Spec ]
```

ما يتبع "WFC" هو اسم القيد المصاغ جيداً وهو "Unique Att Spec" فـــي هـــذا المثـــال.

وعموماً، إذا نظرت قليلاً في الإنتاج، ستجد القيد الذي يحتوي على الاسم الموجود.على سبيل المثال، يؤدي النظر في Production [40] إلى اكتشاف ما يلي:

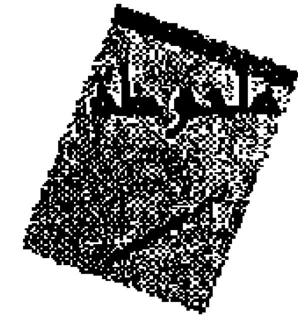
Well-formedness Constraint: Unique Att Spec

No attribute name may appear more than once in the same start tag or empty-element tag.

قد لا يظهر اسم سمة أكثر من مرة في نفس علامة البدء أو علامة العنصر الفارغ. مما يعني أن السمة المتاحة قد لا تظهر أكثر من مرة في العنصر الفردي. فمثلاً تتعدى العلامة التالية الصياغة الجيدة.

```
<P COLOR="red" COLOR="blue">
```

يتم استخدام القيود المصاغة بشكل جيد للمتطلبات الشبيهة بذلك التي يصعب أو يستحيل وضعها في شكل التدقيق النحوي BNF. وكما تقوم توزيعات XML بقراءة المستند، يجب ألا تقوم بالتأكد من تطابق المستند مع إنتاج المستند للتدقيق النحوي BNF ولكن يجب أيضاً التأكد من وفائه لجميع قيود الصياغة الجيدة.



يوجد أيضاً قيود التحقق من الصحة التي يجب الوفاء بها بواسطة المستندات الصحيحة لا يتطلب التحقق من قيود التحقق من الصحة إلى معالجات XML. وتعمل معظم قيود التحقق من الصحة مع الإعلانات الموجودة في DTD سيتم تغطية قيود التحقق من الصحة فيما بعد في هذا الملحق.

الإنتاج المرفق بالقيود الجيدة الصياغة

يقوم هذا المقطع بسرد الإنتاج المرفق بالقيود الجيدة الصياغة وبتوضيحها. لا يحتوي معظم الإنتاج على قيود جيدة الصياغة لذلك لم يتم سرد معظم الإنتاج هنا حيث أن قائمة الإنتاج الكاملة توجد في جزء BNF Grammar من هذا الملحق.

[29] markupdecl ::= elementdecl | AttlistDecl | EntityDecl | NotationDecl | PI | Comment
[Well-formedness Constraint: PEs in Internal Subset]

يوضح القيد المصاغ جيداً أنه لا يمكن استخدام مراجع كينونة المعامل المعرفة في المجموعة الجزئية الداخلية DTD في إعلان لغة الترميز. على سبيل المثال، يعتبر ما يلي غير صحيح:

```
<!ENTITY % INLINES SYSTEM "(I | EM | B | STRONG | CODE)*">
```

```
<!ELEMENT P %INLINES; >
```

ولكنه من ناحية أخرى، يعتبر صحيحاً في المجموعة الجزئية الخارجية DTD.

[39] element ::= EmptyElemTag | STag content ETag

[Well-Formedness Constraint: Element Type Match]

يوضح القيد المصاغ بشكل جيد أنه من المفترض أن يطابق البدء اسم علامة الانتهاء المقابلة لها. ويعتبر ما يلي مثال على العناصر المصاغة بشكل جيد:

```
<TEST>content</TEST>
<test>content</test>
```

بينما لا يعتبر ما يلي كذلك:

```
<TEST>content</test>
<Fred>content</Ethel>
```

[40] STag ::= '<' Name (S Attribute)* S? '>'

[Well-formedness Constraint: Unique Att Spec]

يوضح هذا القيد أن السمة المتاحة قد لا تظهر أكثر من مرة في العنصر الفردي. على سبيل المثال، تتجاوز العلامات التالية الصياغة الجيدة: ،

```
<P COLOR="red" COLOR="blue">
<P COLOR="red" COLOR="red">
```

تعتبر المشكلة هي ظهور سمة COLOR مرتين في نفس العلامة. ولديهم في الحالة الثانية تكرار نفس القيمة مرتين حيث أنها لا تزال مصاغة بشكل سيئ. تعتبر كل من العلامات التالية مصاغة بشكل جيد نظراً لاحتواء السمات على أسماء مختلفة:

```
<P COLOR1="red" COLOR2="blue">
<P COLOR1="red" COLOR2="red">
```

[41] Attribute ::= Name Eq AttValue

[Well-formedness Constraint: No External Entity References]

يوضح هذا القيد أن قيم السمات قد لا تحتوي على مراجع الكينونة التي تشير إلى البيانات الموجودة في المستندات الأخرى. فمثلاً انظر إلى هذه السمة:

```
<BOX COLOR="&RED;" />
```

يعتمد شكل الصياغة الجيد لهذه السمة على كيفية تعريف الكينونة RED. فإذا تم تعريفها بالكامل في DTD سواء في المجموعة الجزئية الداخلية أو الخارجية، تعتبر هذه العلامة مصرح بها، مثل:

```
<!ENTITY RED "#FF0000">
```

ولكن إذا تم تعريف الكينونة RED ككينونة خارجية تشير إلى ملف منفصل، لن يعتـــبر هــذا التعريف جيد. وفي هذه الحالة، سيبدو إعلان ENTITY كما يلي:

```
<!ENTITY RED SYSTEM "red.txt" NDATA COLOR>
```

لاحظ تطبيق هذا القيد على الكينونات الموزعة وعدم تطبيقها إلى الكينونات الأخــــرى غــير الموزعة المتاحة كقيمة سمة من نوع ENTITY أو ENTITIES. على سبيل المثال، يعتبر ما يلي صحيحاً بالرغم من اعتبار RED كينونة خارجية يتم استخدامها كقيمة سمة.

```
<?xml version="1.0"?>
<!DOCTYPE EXAMPLE [
  <!ELEMENT EXAMPLE ANY>
  <!NOTATION COLOR SYSTEM "x-color">
  <!ENTITY RED SYSTEM "red.txt" NDATA COLOR>
  <!ATTLIST EXAMPLE HUE ENTITY #REQUIRED>
]>
<EXAMPLE HUE="RED">
testing 1 2 3
</EXAMPLE>
```

[Well-formedness Constraint: No < in Attribute Values]

يعتبر هذا القيد بسيط للغاية فلا يمكن اعتبار علامة أصغر من (<) جزء من قيمة السمة على سبيل المثال، تعتبر العلامات التالية غير صحيحة:

```
<BOX COLOR="<6699 FF>" />
<HALFPLANE REGION="X < 8" />
```

يمنع [10] Production هذه العلامات فنياً فالغرض الحقيقي من هذا القيد هو التأكد من أنــه لا يمكن إدراج علامة (<) في مرجع الكينونة الخارجي ويعتبر استخدام مرجع الكينونة < مثل ما يلي هو الطريقة الصحيحة لإدراج (<) في قيمة سمة:

```
<BOX COLOR="&lt ; 6699 FF>" />
<HALFPLANE REGION="X &lt; 8" />
```

[44] EmptyElemTag ::= '<' Name (S Attribute)* S? '/>'

[Well-formedness Constraint: Unique Att Spec]

يوضح هذا القيد أن السمة المتاحة قد لا تظهر أكثر من مرة في العنصر الفارغ الفردي. وفيما يلي العلامات التي تتعدى الصياغة الجيدة.

<P COLOR="red" COLOR="blue" />

<P COLOR="red" COLOR="red" />

انظر إلى المثال الثاني. وستجد أن السمة الإضافية تتعدى الصياغة الجيدة.

[60] DefaultDecl ::= '#REQUIRED' | '#IMPLIED' | (('#FIXED' S)? AttValue)

[Well-formedness Constraint: No < in Attribute Values]

يعتبر ذلك هو نفس القيد الموضح في Production [41]. مما يوضح أنه لا يمكن وضع (<) في قيمة سمة افتراضية في إعلان <!ATTLIST> وفيما يلي إعلانات السمة السيئة الصياغة:

<!ATTLIST RECTANGLE COLOR CDATA "<330033>">

<!ATTLIST HALFPLANE REGION CDATA "X < 0" />

[66] CharRef ::= '&#' [0-9]+ ';' | '&#x' [0-9a-fA-F]+ ';'

[Well-formedness Constraint: Legal Character]

يوضح هذا القيد أنه من المفترض صحة الأحرف المشار إليها بواسطة مراجع الأحرف إذا تم كتابتها في المستند. تعتبر مراجع الأحرف مناسبة لإدخال الأحرف الصحيحة التي يصعب كتابتها في نظام معين ولكنها ليست وسيلة لإدخال أحرف ممنوعة.

يتيح Production [2] تعريف الحرف الصحيح:

[2] Char ::= #x9 | #xA | #xD | [#x20-#xD7FF]
| [#xE000-#xFFFD] | [#x10000-#x10FFFF]

تعتبر العناصر الأساسية الهامة هنا هي الحرف التي لم يتم تضمينها وهي بالتحديد أحرف عناصر تحكم ASCII غير القابلة للطباعة والتي يعتبر الشائع منها هو كل من bell و vertical tab و formfeed وsurrogates من #xD800 إلى #xDFFF و#xFFFE non-character.

[68] EntityRef ::= '&' Name ';'

[Well-formedness Constraint: Entity Declared]

تعتبر الغاية من هذا القيد المصاغ جيداً هو التأكد من إعلان جميع الكينونات المستخدمة في المستند DTD باستخدام <!ENTITY>. ومع ذلك يوجد (two loopholes) فتحتان في الحلقة:

١- لا تحتاج الكينونات الخمس المعرفة مسبقاً وهي < و' و", > و&amp إلى إعلان بالرغم من أنه قد يتم الإعلان عنها.

٢- يمكن أن يسمح معالج غير صحيح بكينونات غير معلن عنها التي يمكن أن يتم الإعلان عنها في المجموعة الجزئية الخارجية DTD "التي لا تحتاج إلى معالج غير صحيح لقراءتها". ويمكن إعلان الكينونات في المجموعة الجزئية الخارجية DTD إذا:

أ- لا يحتوي إعلان المستند المستقل بذاته على "standalone="yes.

ب- تحتوي DTD على مرجع كينونة معامل واحد على الأقل.

إذا تم تجاوز أي من هذه الشروط، لن يتم السماح بكينونات غير معلنة "غير الخمسة المتاحـة في loophole one".

يحدد أيضاً هذا القيم أنه إذا تمَ إعلان الكينونات يجب أن يتم إعلانها قبل استخدامها.

[Well-formedness Constraint: Parsed Entity]

يوضح هذا القيد أن مراجع الكينونات قد تحتوي فقط على أسماء الكينونات الموزعة. تتواجـد أسماء الكينونات الموزعة فقط في قيم السمات من نوع ENTITY أو ENTITIES. وكمثال علـى ذلك يعتبر ما يلي مستند سيئ الصياغة.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
 <!ELEMENT DOCUMENT ANY>
 <!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
   NDATA GIF>
<!NOTATION GIF SYSTEM "image/gif">
]>
<DOCUMENT>
 &LOGO;
</DOCUMENT>
```

يعتبر ما يلي هو الطريقة الصحيحة لإدراج كينونة LOGO غير الموزعة في المستند:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [

 <!ELEMENT DOCUMENT ANY>
 <!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
   NDATA GIF>
 <!NOTATION GIF SYSTEM "image/gif">
 <!ELEMENT IMAGE EMPTY>
 <!ATTLIST IMAGE SOURCE ENTITY #REQUIRED>

]>
```

```
<DOCUMENT>
  <IMAGE SOURCE="LOGO" />
</DOCUMENT>
```

[Well-formedness Constraint: No Recursion]

يوضح القيد المصاغ جيداً أنه لا يمكن لكينونة موزعة الإشارة إلى نفسها. على سبيل المثـال، يعتبر this open source classic سيجيء الصياغة.

```
<!ENTITY GNU "&GNU;'s not Unix!">
```

تعتبر المراجع الدائرية أيضاً غير صحيحة:

```
<!ENTITY LEFT  "Left &RIGHT; Left!">
<!ENTITY RIGHT "Right &LEFT; Right!">
```

لاحظ أن التكرار فقط هو ما يعتبر سيئ الصياغة وليس الاستخدام المجرد لمرجـــع كينونـــة واحد داخل الآخر يعتبر ما يلي جيد نظراً لأنه بالرغم من اعتماد كينونة COPY99 على كينونـــة ERH، لا تعتمد كينونة ERH على كينونة COPY99.

```
<!ENTITY ERH    "Elliotte Rusty Harold">
<!ENTITY COPY99 "Copyright 1999 &ERH;">
```

[69] PEReference ::= '%' Name ';'

[Well-formedness Constraint: No Recursion]

يعتبر ذلك هو نفس القيد الذي تم تطبيقه على [68] Production. لا يمكن تكرار كينونـــات المعامل أكثر من الكينونات العامة. على سبيل المثال، يعتبر أيضاً إعلان هـــذه الكينونـــة ســـيئ الصياغة.

```
<!ENTITY % GNU "%GNU;'s not Unix!">
And this is still illegal:
<!ENTITY % LEFT  "Left  %RIGHT; Left!">
<!ENTITY % RIGHT "Right %LEFT; Right!">
```

[Well-formedness Constraint: In DTD]

يشترط ذلك القيد المصاغ جيداً إظهار مراجع كينونة المعامل فقط في DTD. وقد لا تظهر في محتوى المستند أو في أي مكان آخر غير DTD.

قيود التحقق من الصحة

يتم تصميم موضوع المرجع للمساعدة على فهم المطلوب لتحقيق صحة مستند XML. وغالباً مـــا يتم الاستفادة من التحقق من الصحة بالرغم من أنها غير مطلوبة. يمكنك تحقيق الكثير باســـتخدام

المستندات المصاغة بشكل جيد والتي غالباً ما يمكن كتابتها بسهولة نظراً لإتاحـــة القليـل مـن القواعد المفترض اتباعها. ومن المفترض بالنسبة للمستندات الصحيحة، اتباع التدقيـــق النحـوي BNF والقيود المصاغة بشكل جيد وقيود التحقق من الصحة الموضحة في هذا المقطع.

ما هو قيد التحقق من الصحة؟

يعتبر قيد التحقق من الصحة هو قاعدة يجب على المستند الصحيح الالتزام بهاز لا يعتبر جميــع مستندات XML صحيحة وليس بالضرورة أن يتسبب حدوث خطأ في المستند فـــي تحقيـق قيـد التحقق من الصحة. تحتوي معالجات التحقق من الصحة على خيار تمرير التجاوزات الخاصــة بهذه القيود كأخطاء ولكنها ليس بحاجة إلى ذلك ومع ذلك، من المفترض تقرير جميع تجـــاوزات أخطاء بناء جمل "BNF" والصياغة الجيدة.

يمكن التحقق من صحة المستندات التي تحتوي على DTDs فقط.

قيود التحقق من الصحة في XML 1.0:

يقوم هذا المقطع بسرد وتوضيح جميع قيود التحقق من الصحة في قياس XML 1.0. يتم ترتيــب هذه القيود وفقاً لقاعدة BNF التي ينطبق عليها كل قيد.

[28] doctypedecl ::= '<!DOCTYPE' S Name (S ExternalID)? S? ('[' (markupdecl | PEReference | S)* ']' S?)? '>'

Validity Constraint: Root Element Type

يوضح هذا القيد أن الاسم المتاح في إعلان DOCTYPE يجب أن يطابق العنصر الجذري أي أنه من المفترض أن يكون ما يلي متشابه.

```
<?xml version="1.0"?>
<!DOCTYPE ROOTNAME [
  <!ELEMENT ROOTNAME ANY>
]>
<ROOTNAME>
  content
</ROOTNAME>
```

يعتبر أيضاً إعلان العنصر الجذري شيئاً صحيحاً ويحدث ذلك من خلال الســـطر لمكتــوب بالخط المائل ومع ذلك، يتم طلب هذا الإعلان بواسطة قيد تحقق من صحة مختلف وليـــس هــذا القيد.

[29] markupdecl ::= elementdecl | AttlistDecl | EntityDecl | NotationDecl | PI | Comment

Validity Constraint: Proper Declaration/PE Nesting

يشترط هذا القيد أيضاً أن يحتوي إعلان لغة الترميز على كينونة معامل أو أكثر لن يتم تقسيمها على كينونة معامل. فعلى سبيل المثال، تذكر إعلان هذا العنصر.

```
<!ELEMENT PARENT ( FATHER | MOTHER )>
```

تعتبر كينونة المعامل التي تم الإعلان عنها بواسطة إعلان الكينونة التالي هو البديل الصحيح لنموذج المحتوى نظراً لاحتواء كينونة المعامل على كل من (>) و(<) :

```
<!ENTITY % PARENT_DECL "<!ELEMENT PARENT ( FATHER | MOTHER )>">
```

وبإتاحة هذه الكينونة، يمكن إعادة كتابة إعلان العنصر كما يلي:

```
%PARENT_DECL;
```

يعتبر ذلك صحيحاً نظراً لاحتواء كينونة المعامل على كل من (>) و(<) أما الخيار الآخر هو تضمين جزء واحد فقط من إعلان العنصر في كينونة المعامل. على سبيل المثال، إذا توفر لديك العديد من العناصر التي تحتوي على نموذج المحتوى (FATHER | MOTHER)، قد يكون من الأفضل القيام بشيء مثل ما يلي:

```
<!ENTITY % PARENT_TYPES "( FATHER | MOTHER )">
<!ELEMENT PARENT %PARENT_TYPES;>
```

لم يتم تضمين (> أو <) في كينونة المعامل لا يمكن وضع أحد أقواس الزاوية في كينونة المعامل بدون تضمين مساعدها. على سبيل المثال، لا يعتبر ما يلي صحيحاً حتى إذا ظهر ليتوسع في إعلان عنصر صحيح.

```
<!ENTITY % PARENT_TYPES "( FATHER | MOTHER )>">
<!ELEMENT PARENT %PARENT_TYPES;
```

لاحظ إن المشكلة لا تكن في احتواء نص استبدال كينونة المعامل على حرف < حيث أن ذلك يعتبر صحيحاً (بخلاف استخدام حرف > الذي قد يعتبر غير صحيحاً في إعلان كينونة المعامل الداخلي). تعتبر المشكلة هي كيفية استخدام < لإنهاء إعلان عنصر بدأ في كينونة أخرى.

[32] SDDecl ::= S 'standalone' Eq (("'" ('yes' | 'no') "'") | ('"' ('yes' | 'no') '"'))

Validity Constraint: Standalone Document Declaration

يوضح هذا القيد باختصار أن المستند يجب أن يحتوي على إعلان مستند مستقل بذاته يحتوي على القيمة no (standalone="no") إذا تطلب تشغيل الملف وتحديد التحقق من صحته أي

ملفات أخرى يؤثر ذلك في الغالب على مجموعات DTD الجزئية الخارجية المرتبطة. بكينونـات المعامل.

- تتوفر هذه الحالة إذا كان ما يلي صحيحاً:
- يتم الإعلان عن الكينونة المستخدمة في المستند في المجموعة الجزئية الخارجية DTD.
- توفر المجموعة الجزئية الخارجية DTD القيم الافتراضية للسمات التي تظهر في المستند بدون قيم.
- تقوم المجموعة الجزئية الخارجية DTD بتغيير كيفية تطبيع قيم السمات في المستند.
- تعلن المجموعة الجزئية DTD الخارجية عن العناصر التي تعتبر توابعها مجرد عناصر "ليست بيانات أحرف أو محتوى مختلط" عند احتواء نفس هذه التوابـع علـى مسافة بيضاء.

[39] element ::= EmptyElemTag | STag content ETag
Validity Constraint: Element Valid

يوضح هذا القيد أن هذا العنصر لا يطابق إعلان العنصر الموجود في DTD. لذلك يجـب أن يكون أحد ما يلي صحيحاً:

١- ألا يحتوي العنصر على محتوى وأن يعلن إعلان العنصر من عنصر EMPTY.

٢- يحتوي العنصر فقط على عناصر تابعة تتطابق مع التعبير المنتظم الموجود في نمـوذج محتوى العنصر.

٣- يتم الإعلان عن العنصر باعتباره يحتوي على محتوى مختلط كمـا يحتـوي محتـوى العنصر على بيانات أحرف وعناصر تابعة يتم إعلانها في إعلان المحتوى المختلط.

٤- يتم الإعلان عن العنصر باعتباره ANY وكما يتم الإعلان عن جميع العناصر التابعة.

[41] Attribute ::= Name Eq AttValue
Validity Constraint: Attribute Value Type

يوضح هذا القيد أنه يجب الإعلان عن اسم السمة في إعلان ATTLIST في DTD بالإضافـة إلى ذلك، يجب أن تتطابق قيمة السمة مع النوع المعلن في إعلان ATTLIST.

[45] elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'
Validity Constraint: Unique Element Type Declaration

لا يمكن الإعلان عن عنصر أكثر من مرة في DTD سواء كانت الإعلانات متوافقـة أم لا، فعلى سبيل المثال، يعتبر ما يلي صحيحاً.

```
<!ELEMENT EM (#PCDATA)>
```

بينما لا يعتبر ذلك صحيحاً:

```
<!ELEMENT EM (#PCDATA)>
<!ELEMENT EM (#PCDATA | B)>
```

ولا يعتبر ذلك أيضاً صحيحاً:

```
<!ELEMENT EM (#PCDATA)>
<!ELEMENT EM (#PCDATA)>
```

مما قد يتسبب في بعض المشاكل التي تنتج عن دمج مجموعات DTD الجزئية الخارجية من عدة مصادر مختلفة تقوم كلاهما بالإعلان عن بعض العناصر المتشابهة. قد تساعد مسافات الأسماء على حل ذلك ولكن في أضيق الحدود.

[49] choice ::= '(' S? cp (S? '|' S? cp)* S? ')'
Validity Constraint: Proper Group/PE Nesting

يوضح هذا القيد أن الخيار قد يحتوي أو يتم احتواؤه في واحد أو اكثر من كينونات المعامل ولكن قد لا يتم تقسيمه في كينونة معامل انظر إلى إعلان هذا عنصر على سبيل المثال:

```
<!ELEMENT PARENT ( FATHER | MOTHER )>
```

تعتبر كينونة المعامل التي تم الإعلان عنها بواسطة إعلان الكينونة التالي هي البديل الصحيح لنموذج المحتوى نظراً لاحتواء كينونة المعامل على كل من "و".

```
<!ENTITY % PARENT_TYPES "( FATHER | MOTHER )">
```

يمكنك إعادة كتابة إعلان العنصر كما يبدو:

```
<!ELEMENT PARENT %PARENT_TYPES;>
```

يعتبر ذلك صحيحاً. نظراً لاحتواء كينونة المعامل على كل من "و" أما الخيار الأخر هو تضمين العناصر التابعة فقط والإبقاء على كلا القوسين مثل:

```
<!ENTITY % PARENT_TYPES " FATHER | MOTHER ">
<!ELEMENT PARENT ( %PARENT_TYPES; )>
```

تعتبر الميزة هنا هي أنه بإمكانك إضافة عناصر إضافية لم يتم تعريفها في كينونة المعامل مثل:

```
<!ELEMENT PARENT ( UNKNOWN | %PARENT_TYPES; ) >
```

ولكن ما لا يمكن تنفيذه هو وضع أحد الأقواس في كينونة المعامل بدون تضمين مرادفه.
يعتبر ما يلي غير صحيح بالرغم من ظهوره ليتوسع في إعلان عنصر صحيح.

```
<!ENTITY % FATHER " FATHER )">
```

```
<!ENTITY % MOTHER " ( MOTHER | ">
<!ELEMENT PARENT %FATHER; %MOTHER; ) >
```

تعتبر مشكلة هذا المثال هي إعلان ELEMENT وليست إعلانات ENTITY. قد يصح إعلان الكينونات كما حدث هنا ولكن يتسبب استخدامه في مضمون خيار في عدم صحته.

[50] seq ::= '(' S? cp (S? ',' S? cp)* S? ')'

Validity Constraint: Proper Group/PE Nesting

يعتبر ذلك هو نفس القيد كتلك الموجود بأعلى باستثناء أنه يتم تطبيقه عل التسلسلات بدلاً من الخيارات يتطلب ذلك احتواء التسلسل على كينونة معامل أو أكثر ولكن قد لا يتم تقسيمه عبر كينونة معامل. وكمثال، انظر إلى إعلان هذا العنصر:

```
<!ELEMENT ADDRESS ( NAME, STREET, CITY, STATE, ZIP )>
```

تعتبر كينونة المعامل التي تم إعلانها بواسطة إعلان الكينونة التالي هي البديل الصحيح لنموذج المحتوى نظراً لاحتواء كينونة المعامل على كل من "و".

```
<!ENTITY % SIMPLE_ADDRESS "( NAME, STREET, CITY, STATE, ZIP )">
```

يمكنك كتابة إعلان العنصر كما يلي:

```
<!ELEMENT ADDRESS %SIMPLE_ADDRESS;>
```

يعتبر ذلك صحيحاً نظراً لاحتواء كينونة المعامل على كل من "و" ويعتبر الخيار الآخر هو تضمين العناصر التابعة فقط والإبقاء على كلا القوسين، مثل:

```
<!ENTITY % SIMPLE_ADDRESS " NAME, STREET, CITY, STATE, ZIP ">
<!ELEMENT ADDRESS( %SIMPLE_ADDRESS; )>
```

تعتبر الميزة وراء ذلك هي إمكانية إضافة العناصر الإضافية التي لم يتم تعرفها في كينونة المعامل، مثل:

```
<!ENTITY % INTERNATIONAL_ADDRESS " NAME, STREET, CITY,
  PROVINCE?, POSTAL_CODE?, COUNTRY ">
<!ELEMENT ADDRESS ( (%SIMPLE_ADDRESS;)
          | (%INTERNATIONAL_ADDRESS;) ) >
```

ولكن يستحيل وضع أحد الأقواس في كينونة المعامل بدون تضمين مرادفه لا يعتبر ما يلي صحيحاً، بالرغم من ظهوره في إعلان العنصر الصحيح:

```
<!ENTITY % SIMPLE_ADDRESS_1 "( NAME, STREET, ">
<!ENTITY % SIMPLE_ADDRESS_2 "CITY, STATE, ZIP)">
```

```
<!ELEMENT ADDRESS %SIMPLE_ADDRESS_1; %SIMPLE_ADDRESS_2; )
>
```

تعتبر المشكلة في هذا المثال هي إعلان ELEMENT وليس إعلانات ENTITY. يعتبر إعلان كينونات مثل تلك السابقة صحيحاً ولكن يتسبب استخدامها في مضمون تسلسل في عدم صحتها:

[51] Mixed ::= '(' S? '#PCDATA' (S? '|' S? Name)* S? ')*' | '(' S? '#PCDATA' S? ')'

Validity Constraint: Proper Group/PE Nesting

يعتبر ذلك نفس القيد تماماً ولمنه الآن يتم تطبيقه على المحتوى المختلط بدلاً من الاختيـارات أو التسلسلات. مما يشترط احتواء نموذج محتوى مختلط في كينونة معامل ولكن قد لا يتم تقسيمه في كينونة معامل. وكمثال، انظر إعلان هذا العنصر:

```
<!ELEMENT P ( #PCDATA | I | EM | B | STRONG )>
```

تعتبر كينونة المعامل التي تم الإعلان عنها بواسطة إعلان الكينونة التالي هو بديل صحيـح لنموذج المحتوى نظراً لاحتواء كينونة المعامل على كل من "و":

```
<!ENTITY % INLINES "( #PCDATA | I | EM | B | STRONG )">
```

يمكن إعادة كتابة إعلان العنصر كما يلي:

```
<!ELEMENT P %INLINES;>
```

يعتبر ذلك صحيحاً نظراً لاحتواء كينونة المعامل على كل من "و" ويعتبر الخيار الآخر هـو تضمين أجزاء المحتوى فقط ولكن مع الإبقاء على كلا القوسين، مثل:

```
<!ENTITY % INLINES " #PCDATA | I | EM | B | STRONG ">
<!ELEMENT P ( %INLINES; ) >
```

تعتبر الميزة هنا هي إمكانية إضافة عناصر إضافية لم يتم تعريفها في كينونة المعامل، مثل:

```
<!ELEMENT QUOTE ( %INLINES; | SPEAKER ) >
```

ومع ذلك، يستحيل وضع أحد القواس في كينونة المعامل بدون تضمين مرادفه. لذلك لا يعتبر ما يلي، صحيح بالرغم من ظهوره في إعلان العنصر الصحيح:

```
<!ENTITY % INLINES1 " I | EM | B | STRONG )">
<!ENTITY % INLINES2 " ( #PCDATA | SPEAKER | ">
<!ELEMENT QUOTE %INLINES1; %INLINES2; ) >
```

تعتبر المشكلة في هذا المثال هي إعلان ELEMENT وليسـت إعلانـات ENTITY. يعتـبر إعلان الكينونات كما تم هنا عملاً صحيحاً ولكن يتسبب استخدامها فـي مضمـون اختيـار "أو تسلسل" في عدم صحتها.

Validity Constraint: No Duplicate Types

لا يمكن تكرار أي عنصر في إعلان المحتوى المختلط وكمثال، ما يلي غير صحيح:

```
( #PCDATA | I | EM | I | EM )
```

لا يوجد ما يدعو إلى كتابة إعلان محتوى مختلط مثل هذا ولكن في نفس الوقـــت لا يعتـبر الضرر واضحاً. تتيح الاختبارات المجردة نماذج محتوى مثلما يلي:

```
( I | EM | I | EM )
```

تظهر مشكلة عند خلط #PCDATA.



يعتبر هذا الاختيار غامض حيث أنه عند مواجهة الموزع لأي مـــن I أو EM، لا يتمكن الاختيار من معرفة مطابقته للنموذج الأول أو الثاني في نموذج المحتـــوى لذلك بالرغم من صحته، تقوم بعض الموزعات بتقريره كخطأ يجب تجنبه إذا كان ممكناً.

[56] TokenizedType ::= 'ID' | 'IDREF' | 'IDREFS' | 'ENTITY' | 'ENTITIES' | 'NMTOKEN' | 'NMTOKENS'

Validity Constraint: ID

يجب أن تكون قيم السمات من نوع ID أسماء XML صحيحة (Production [5]). بالإضافة إلى أنه لا يمكن استخدام اسم فردي أكثر من مرة في نفس المستند كقيمة سمة نوع ID باعتبـــاره ID "رقم معرف":

```
<BOX ID="B1" WIDTH="50" HEIGHT="50" />
<BOX ID="B1" WIDTH="250" HEIGHT="250" />
```

يعتبر ذلك أيضاً غير صحيح نظراً لعدم إمكانية أسماء XML على البدء بأرقام.

```
<BOX ID="1" WIDTH="50" HEIGHT="50" />
```

يعتبر ذلك صحيحاً إذا لم يحتوي NAME على نوع ID:

```
<BOX ID="B1" WIDTH="50" HEIGHT="50" />
<BOX NAME="B1" WIDTH="250" HEIGHT="250" />
```

من ناحية أخرى، يعتبر هذا المثال غير صحيح إذا لم يحتوي NAME على نوع ID وحتى إذا كانت سمة NAME مختلفة عن سمة ID. بالإضافة إلى أن ما يلي لا يعتبر صحيحاً إذا احتـــوى على نوع ID حتى إذا كان يوجد عنصرين مختلفين:

```
<BOX NAME="FRED" WIDTH="50" HEIGHT="50" />
```

```
<PERSON NAME="FRED" />
```

من المفترض أن تكون قيم السمة ID فريدة عبر جميع سمات العناصر وID وليس فقط فئة معينة من أو سمات معينة لفئة معينة من العناصر.

Validity Constraint: One ID per Element Type

يحتوي كل عنصر على أحد سمات النوع ID. على سبيل المثال، يعتبر ما يلي غير صحيح:

```
<!ELEMENT PERSON (ANY) >
<!ATTLIST PERSON SS_NUMBER   ID #REQUIRED>
<!ATTLIST PERSON EMPLOYEE_ID ID #REQUIRED>
```

Validity Constraint: ID Attribute Default

يجب إعلان جميع سمات النوع ID باعتبارهـا #IMPLIED أو #REQUIRED حيث أن #FIXED غير متاحة وكمثال على ذلك، يعتبر ما يلي غير صحيح:

```
<!ATTLIST PERSON SS_NUMBER ID #FIXED "SS123-45-6789">
```

تعتبر المشكلة هي إذا توفر أكثر من عنصر PERSON في المستند، سيتم تجاوز قيد التحقق من الصحة ID آلياً.

Validity Constraint: IDREF

يحدد قيد التحقق من الصحة IDREF أنه من المفترض أن تكون قيمة السمة لنوع السمة IDREF هو نفس قيمة سمة نوع ID لعنصر موجود في المستند قد تشير سمات IDREF المتعددة الموجودة في نفس العناصر أو في العناصر المختلفة إلى عنصر فردي يجب أن تكون قيم سمات ID فريدة "على الأقل ضمن قيم سمات ID الأخرى في نفس المستند" ولكن لا تشترط سمات IDREF أن تكون فريدة.

بالإضافة إلى ذلك، يجب أن تكون قيم سمات النوع IDREFS قائمة بقيم سمات ID مفصولة بمسافة بيضاء عن العناصر الموجودة في المستند.

Validity Constraint: Entity Name

يجب إن تكون قيمة السمة التي تعتبر ENTITY هي نوعها المعلن عنه هو اسم كينونة "non-parameter" عام غير موزع تم الإعلان عنه في DTD سواء في المجموعة الجزئية الداخلية أو الخارجية.

يجب أن تكون قيمة السمة التي يعتبر نوعها المعلن عنه هـي ENTITIES قائمة بأسماء كينونات "non-parameter" العامة غير الموزعة المفصولة بمسافة بيضاء تم الإعلان عنها في DTD سواء في المجموعة الجزئية الداخلية أو الخارجية.

Validity Constraint: Name Token

يجب إن نتطابق قيمة السمة التي يعتبر نوعها المعلن عنـــه هــو NMTOKEN إنتــاج XML (Production [7]) NMTOKEN ممال يعني أنه يجب إن يتكون من حرف اسم واحد أو اكــثر أو نقطة أو واصلة وحرف ربط أو امتداد.

يجب أن تكون قيمة السمة التي يعتبر نوعها المعلن عنه هو NMTOKENS قائمـــة برمــوز أسماء مفصولة بمسافة بيضاء. وكمثال على ذلك، يعتبر ما يلي عنصـــر يحتــوي علـــى ســمة COLORS من نوع NMTOKENS:

```
<BOX WIDTH="50" HEIGHT="50" COLORS="red green blue" />
```

يعتبر ما يلي عنصر غير صحيح يحتوي على سمة COLORS من نوع NMTOKENS:

```
<BOX WIDTH="50" HEIGHT="50" COLORS="red, green, blue" />
```

[58] NotationType ::= 'NOTATION' S '(' S? Name (S? '|' S? Name)* S? ')'

Validity Constraint: Notation Attributes

يجب أن تكون قيمة السمة التي يعتبر نوعها المعلن عنه هو NOTATION اســم رمــز تــم الإعلان عنه في DTD.

[59] Enumeration ::= '(' S? Nmtoken (S? '|' S? Nmtoken)* S? ')'

Validity Constraint: Enumeration

يجب أن تكون قيمة السمة التي يعتبر نوعها المعلن عنه هو ENUMERATION قائمة برموز الأسماء المفصولة بمسافة بيضاء. لا يجب أن تكون هذه الأسماء بالضرورة أسماء أي شيء معلن عنه في DTD أو في أي مكان آخر. يجب أن تتطابق مـــع إنتـــاج NMTOKEN (Production [7]). يعتبر ما يلي تعدد غير صحيح نظراً لاستخدام الفواصل. بدلاً من المسافة البيضاء لفصـــل رموز الأسماء:

```
( red, green, blue)
```

يعتبر ذلك تعدد غير صحيح نظراً لوضع رموز الأسماء في علامات اقتباس:

```
( "red" "green" "blue")
```

لا تعتبر أي من الفواصل لا توجد إمكانية لسوء تفسير هذه الخطاء الشائعة باعتبارها قائمـــة برموز أسماء غير عادية مفصولة بمسافة بيضاء.

[60] DefaultDecl ::= '#REQUIRED' | '#IMPLIED' | (('#FIXED' S)? AttValue)

Validity Constraint: Required Attribute

إذا تم الإعلان عن سمة عنصر باعتبارها #REQUIRED فيعتبر ذلك خطأً في التحقق مـــن صحة أي نموذج عنصر لكي لا يتم إتاحة أي قيمة لهذه السمة.

Validity Constraint: Attribute Default Legal

يوضح قيد التحقق من الصحة أن أي قيمة سمة افتراضية متاحة في إعلان ATTLIST يجــب أن تفي بقيود السمة من هذا النوع. فعلى سبيل المثال، قد لا يعتبر ما يلي صحيحــاً نظــراً لأن القيمة الافتراضية UNKNOWN لا تعتبر أحد الاختيارات المتاحة بواسطة نموذج المحتوى:

```
<!ATTLIST CIRCLE VISIBLE (TRUE | FALSE) "UNKNOWN">
```

قد يعتبر UNKNOWN غير صحيحاً لهذه السمة سواء تم إتاحتها كقيمة افتراضيــة أو فــي عنصر حقيقي مثلما يلي:

```
<CIRCLE VISIBLE="UNKNOWN" />
```

Validity Constraint: Fixed Attribute Default

يوضح قيد التحقق من الصحة الشائع أنه إذا تم الإعلان عن #FIXED في إعلان ATTLIST سوف يقدم ذلك الإعلان قيمة افتراضية. وكمثال، يعتبر ما يلي غير صحيح:

```
<!ATTLIST AUTHOR COMPANY CDATA #FIXED>
```

وفيما يلي إعلان تم تصحيحه:

```
<!ATTLIST AUTHOR COMPANY CDATA #FIXED "TIC">
```

[68] EntityRef ::= '&' Name ';'

Validity Constraint: Entity Declared

يمتد هذا القيد على قيد الصياغة الجيدة لنفس الاسم. يجب أن يتم تعريــف جميــع المراجــع المشار إليها في المستند الصحيح بواسطة إعلانات <!ENTITY> في DTD. يجــب أن تســبق التعريفات أي استخدام للكينونة التي تقوم هذه الإعلانات بتعريفها.

لا تعتبر loophole بالنسبة إلى مستندات التي يتم تطبيقها على المستندات المصاغة بشــكل جيد متاحة ولكن لا تزال loophole الخاصة بالكينونات الخمس المعرفــة مســبقاً وهــم <، &apos و&gt و&quot و&amp متاحة. ومع ذلك من الفضل الإعلان عنهم حتى إذا لم تضطر إلى ذلك. ستبدو هذه الإعلانات كما يلي:

```
<!ENTITY lt   "#&38;#60;">
<!ENTITY gt   #&"62;">
<!ENTITY amp  #&"38;#38;">
<!ENTITY apos #&"39;">
<!ENTITY quot #&"34;">
```

[69] PEReference ::= '%' Name ';'

Validity Constraint: Entity Declared

يعتبر ذلك هو نفس القيد السابق الذي يتم تطبيقه على مراجع كينونات المعامل بدلاً من مراجع الكينونات العامة.

[76] NDataDecl ::= S 'NDATA' S Name
Validity Constraint: Notation Declared

من المفترض أن يكون الاسم المستخدم في إعلان بيانات الرمز (الذي يتـــم اســتخدامه فــي تعريف الكينونة للكينونة غير الموزعة) اسم لرمز يتم الإعلان عنه في OTD. على سبيل المثال، يعتبر المستند التالي صحيحاً ومع ذلك، إذا تم حذف السطر الذي يعلن عن رمز GIF "الموضـــح بالخط العريض" سوف يصبح الرمز غير صحيح.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT ANY>
  <!ENTITY LOGO SYSTEM "http://metalab.unc.edu/xml/logo.gif"
    NDATA gif>
  <!NOTATION GIF SYSTEM "image/gif">
]>
<DOCUMENT>
  &LOGO;
</DOCUMENT>
```

ب

الملحق

مواصفات XML 1.0

يحتوي هذا الملحق على مواصفات XML 1.0 الكاملة والنهائية كما هي منشورة على اتحاد الشبكة الدولية. تمت مراجعة هذا المستند بواسطة أعضاء W3C وجهات أخرى، كما تمت تزكيته بواسطة المدير كتوصية من W3C ويعد هذا المستند متكاملاً ويمكن استخدامه كمادة مرجعية أو دليل معياري من مستند آخر. إذا حدث تغيرات مستقبلية في XML "وهو ما سيحدث بلا شك" فسيتم إصدار رقم جديد إصدارة.

ليس هذا المستند سهل القراءة دائماً، حيث تكون الدقة أكثر أهمية من الوضوح. على أي حال، سيفيدك هذا المستند عندما تريد التأكد من أن المشكلة التي تواجهها في معالج XML أو في رموز XML. وبالتالي سيكون عليك أن تعتاد علية وتكون قادراً على أن تجد فيه ما تريد عندما تحتاج إليه.

قام بكتابة هذا المستند كل من تيم براي وسي إم سبيربرج ماكوين بمساعدة آخرين مذكورين في نهاية هذا المستند.

REC-xml-19980210

هذه الإصدارة:

http://www.w3.org/TR/1998/REC-xml-19980210
http://www.w3.org/TR/1998/REC-xml-19980210.xml
http://www.w3.org/TR/1998/REC-xml-19980210.html
http://www.w3.org/TR/1998/REC-xml-19980210.pdf
http://www.w3.org/TR/1998/REC-xml-19980210.ps

أحدث إصدارة:

http://www.w3.org/TR/REC-xml

الإصدار السابقة:

http://www.w3.org/TR/PR-xml-971208

المحررون:

Tim Bray (Textuality and Netscape) <tssbray@textuality.com>
Jean Paoli (Microsoft) <jeanpa@microsoft.com>

<jeanpa@microsoft.com> C.M. Sperberg-McQueen (University of Illinois at Chicago) <cmsmcq@uic.edu>

مطلق

لغة الترميز القابلة للامتداد (XML) هي مجموعة جزئية من SGML التي يتم توصيفها بالتفصيل في هذا المستند. وهدف هذه اللغة تمكين استخدام واستلام ومعالجة SGML العامة على الويب بنفس طريقة التعامل مع HTML الآن تم تصميم XML لتسهل التنفيذ والعمل المتبادل مع كل من SGML وHTML

حالة هذا المستند

تمت مراجعة هذا المستند بواسطة أعضاء W3C وجهات آخرى، كما تمت تزكيته بواسطة المدير كتوصية من W3C. ويعد هذا المستند متكاملاً ويمكن استخدامه كمادة مرجعية أو دليل معياري من مستند آخر ويتركز دور W3C للتوصية في جذب الانتباه للمواصفات وزيادة توزيعها المنتشر، ما يحسن الأداء والفاعلية على الويب.

يحدد هذا المستند صيغة تم تكوينها بواسطة معيار معالجة النصوص الدولي "لغة الترميز العامة القياسية، ISO 8879:1986(E) حيث تم تعديلها وتصحيحها" ليتم استخدامها على الشبكة الدولية. وهو منتج W3C XML Activity، ويمكن الحصول على تفاصيل بشأنه على: http://www.w3. org/XML كما يمكن العثور على قائمة بتوصيات W3C الحالية بالإضافة إلى مستندات فنية أخرى على: http://www.w3.org/TR.

تستخدم هذه المواصفات URI، المعرف بواسطة [Berners-Lee et al.] في الإنجاز من المتوقع تحديثه [IETF RFC1738] و[IETF RFC1808].

هذه القائمة من الأخطاء المعروفة في هذه المواصفات متاحة على http://www.w3.org/XML/ xml-19980210-errata.

الرجاء الإبلاغ عن الأخطاء في هذا المستند في xml-editor@w3.org.

لغة الترميز القابلة للتحديد (XML) 1.0

جدول المحتويات

١.المقدمة

لغة الترميز القابلة للتحديد "اختصاراً XML" تصف نوعاً من كائنات البيانات تسمى بمستندات XML، كما تصف طريقة تعامل برامج الحاسب التي تقوم بمعالجة هذه المستندات. XML هي تشكيل جانبي لتطبيق أو نموذج محدود SGML وهي لغة الترميز العامة القياسية [ISO 8879] وتطابق مستندات XML من حيث تكوينها مستندات SGML.

يتم تكوين مستندات XML من وحدات تخزين تسمى بالكائنات والتي تحتوي على بيانات موزعة أو غير موزعة تتكون البيانات الموزعة من أحرف يكون بعضها بيانات حرفية وبعضها يكون رموزاً. تكون هذه الرموز تشفيراً لوصف تخطيط تخزين المستند وبنيته المنطقية. توفر XML آلية لفرض قيود على تخطيط التخزين والبنية المنطقية.

هناك وحدة نمطية من البرامج تسمى معالج XML وتستخدم لقراءة مستندات XML وتوفر إمكانية الوصول إلى محتوياتها وأبنيتها. من المفترض أن معالج XML يقوم بعمله بالنيابة عن وحدة نمطية أخرى تسمى التطبيق. تصف هذه المواصفات طريقة عمل معالج XML المطلوبة فيما يتعلق بكيفية الطريقة اللازمة لقراءة بيانات XML والمعلومات التي يجب اللازمة توفيرها للتطبيق.

١-١.الأصول والأهداف

تم تطوير XML بواسطة مجموعة عمل XML (تسمى بمجلس لمراجعة التحريرية) SGML تم تشكيلها تحت رعاية اتحاد الشبكة الدولية (W3C) عام ١٩٩٦.

كانت أهداف تصميم XML هي:

١- يجب أن يكون استخدام XML مباشراً على الإنترنت.

٢- يجب أن تدعم XML العديد والعديد من التطبيقات.

٣- يجب أن تكون XML متوافقة مع SQML.

٤- يجب أن يكون من السهل كتابة البرامج التي تقوم بمعالجة مستندات XML.

٥- يجب أن يكون عدد الميزات الاختيارية في XML أقل ما يكون (نموذجياً لا يكون هناك أي منها).

٦- يجب أن تكون مستندات XML قابلة للقراءة وواضحة منطقياً

٧- يجب أن يكون إعداد تصميم XML سريعاً

٨- يجب أن يكون تصميم XML رسمياً ومختصراً.

٩- يجب أن يكون تكوين مستندات XML سهلاً.

١٠-يكون الاختصار في ترميز XML غير ذي أهمية كبيرة.

توفر هذه المواصفات مع معايير أخرى "Unicode وISO/IEC 10646 الأحرف وإنترنت RFC 1766 لعلامات تعريف اللغات وISO 639 لرموز أسماء اللغات وISO 3166 لرموز أسماء البلاد" جميع المعلومات اللازمة لفهم الإصدارة XML 1.0 ولبناء برامج للحاسب لمعالجتها.

هذه الإصدارة من مواصفات XML يتم توزيعها بحرية طالما جميع الملاحظات النصية والقانونية سليمة.

١-٢. المصطلحات

المصطلحات المستخدمة لوصف مستندات XML يتم تعريفها خلال هذه الموصفات. المصطلحات المعرفة في القائمة التالية يتم استخدامها في بناء هذه التعريفات وفي وصف إجراءات معالج XML.

يمكن (may) للإشارة إلى أن مستندات المطابقة ومعالجات XML مسموحة لكنها لا يجب أن تعمل كما هو واضح.

يجب (must) للإشارة إلى أن مستندات المطابقة ومعالجات XML مطلوبة لكي تعمل كما هو موضح وإلا ستكون مسببة لخطأ.

خطأ (error) بمعنى تجاوز قواعد هذه المواصفات وعدم تحديد النتائج. قد تتبع برامج المطابقة خطأ وتقوم بالتبليغ عنة أو قد تقوم بمعالجة.

خطأ جسيم (fatal error) وهو خطأ يجب أن تتبعه معالج مطابقة XML ويقوم بالتبليغ عنه للتطبيق. بعد العثور على خطأ جسيم، قد يستمر المعالج في معالجة البيانات للبحث عن أخطاء أخرى والتبليغ عنها للتطبيق. لدعم تصحيح الأخطاء، قد يجعل المعالج بعض البيانات غير المعالجة من المستند "مع بيانات حرفية وترميزات متداخلة" متاحة للتطبيق. لمجرد تتبع خطأ جسيم على أي حال يجب ألا يستمر المعالج في المعالجة العادية "بمعنى أنه يجب ألا يستمر في تمرير بيانات ومعلومات حرفية عن البنية المنطقية للمستند للتطبيق بالطريقة العادية". عند اختيار المستخدم "at user name" يجب أو يمكن أن تعمل برامج المطابقة "على حسب فعل الشرط في الجملة" كما هو موضح. إذا كان هذا هو الحال، فيجب أن توفر للمستخدمين طرقاً لتمكين أو تعطيل الأداء الموضح.

قيد الصلاحية (validity constraint) وهو قاعدة تطبيق على جميع مستندات XML الصحيحة. وتعد انتهاكات قيود الصلاحية أخطاء ويجب عند اختيار المستخدم التبليغ عنها بواسطة معالجي XML.

قيد جودة التكوين (well-formedness constraint) وهو قاعدة تنطبق على جميع مستندات XML جيدة التكوين. تعد انتهاكات قيود جودة التكوين أخطاء جسيمة.

مطابقة (match) بين السلاسل أو الأسماء يجب أن يكون اسمين أو سلسلتين يتم مقارنتهما متطابقتين تتطابق الأحرف ذات الأشكال المتعددة في ISO/IEC 10646 "مثل الأحرف ذات نماذج التشكيل سابقة الإعداد والأساسية" إذا كان لها نفس الصورة في كلا السلسلتين عند اختيار المستخدم، يمكن للمعالجات معايرة هذه الأحرف بنموذج مقبول. ولا تتم أي عمليات طي "للسلاسل أو القواعد"، حيث تطابق سلسلة منتجاً قاعدياً إذا كانت تنتمي لنفس اللغة المكونة بواسطة ذلك المنتج. "وللمحتوى وأنواع المحتويات"، حيث يطابق عنصر إعلاناً خاصاً به عندما ينطبق بالطريقة الموضحة في القيد Element Valid.

للتوافقية (for interoperability) وهي خاصية مضافة بشكل منفرد في XML للتأكد من استمرار توافقية XML مع SGML.

للتشغيل المتبادل (for interoperability) وهي إضافة غير إلزامية لزيادة فرص معالجة مستندات XML بواسطة معالجات SGML الأساسية المثبتة بالفعل التي تسبق WebSGML Adaptations Annex أو ملحق تحديثات ISO 8879 WebSGML.

٢. المستندات

كائنات البيانات هي مستندات XML إذا كانت جيدة التكوين، كما هو موضح في هذه المواصفات يمكن لمستند XML جيد التكوين أن يكون صحيحاً أيضاً إذا حقق قيوداً معينة أخرى.

يكون لكل مستند XML بنية منطقية وأخرى حقيقية. البنية الحقيقية هي أن المستند يتكون من وحدات تسمى الكائنات. يمكن أن يشير الكائن إلى كائنات أخرى ليشملها المستند. ويبدأ المستند "جذر" أو كائن مستند. أما منطقياً، فيتكون المستند من إعلانات وعناصر وتعليقات ومراجع أحرف وتعليمات معالجة ويشار إليها جميعاً في المستند بترميز مفصل. يجب تكوين البنية المنطقية والحقيقية بشكل سليم كما هو موضح في "٤-٣-٢ الكائنات الموزعة جيدة التكوين".

٢-١ .مستندات XML جيدة التكوين

يكون الكائن النصي مستند XML جيد التكوين إذاً:

- بشكل كلي، إذا طابق الكائن المستند المعنون المنتج.
- إذا طابق الكائن جميع قيود جودة التكوين المعطاة في هذه المواصفات.

كل كائن موزع مشار إليه في المستند بشكل مباشر أو غير مباشر جيد التكوين.

```
Document
[1] document ::= prolog element Misc*
```

مطابقة منتج المستند يتضمن:

- -أنه يحتوي على عنصر أو أكثر
- أن هناك عنصراً واحداً بالتحديد يسمى "الجذر" root أو عنصر المستند لا يظهر أي جزء منه في محتوى أي عنصر آخر لكل العناصر الأخرى، إذا كانت علامة البداية في محتوى عنصر آخر، تكون علامة النهاية في محتوى نفس العنصر وببساطة، فإن العناصر، المحددة بعلامات البداية وعلامات النهاية، تتداخل بشكل سليم مع بعضها البعض.
- كنتيجة لهذا، لكل عنصر C غير عنصر الجذر، فإن هناك عنصراً آخر P في المستند مثل C في محتوى P، لكنه ليس في محتوى أي عنصر آخر في محتوى P. ويشار إلى P بالأصل parent بالنسبة C، كما يشار إلى C بالتابع P child.

٢-٢ الأحرف

يحتوي الكائن الموزع على نص، أي تسلسل من الأحرف، يمكن أن يمثل ترميزاً أو بيانات حرفية. والحرف هو وحدة دقيقة نصية كما هو معرف بواسطة [ISO/IEC 10646] ISO/IEC 10646 والأحرف الصحيحة هي مفتاح الجولة وحرف الإرجاع والأحرف الرسومية Unicode وISO/IEC 10646. لا ينصح باستخدام "أحرف التوافقية" compatibility characters، كما هي موضحة في قسم 6.8 [Unicode].

```
Character Range
[2] Char ::= #x9 | #xA | #xD | [#x20-#xD7FF]    /* any Unicode
character,
        | [#xE000-#xFFFD]        excluding the surrogate
```

| [#x10000-#x10FFFF]   blocks, FFFE, and

*/

يمكن أن تختلف آلية ترميز نقاط رموز الأحرف إلى أنماط بت من كائن لآخر. يجب أن تقبل جميع معالجات XML ترميزات UTF-8 و UTF-16 10646. تتم مناقشة آليات ترميز أي من الاثنين الذي يتم استخدامه أو استخدام ترميزات أخرى لاحقاً في "٤-٣-٣ ترميز الأحرف في الكائنات".

٢-٣ أبنية تركيبة شائعة

يعرف هذا القسم بعض الرموز المستخدمة كثيراً في القواعد

S (المسافة البيضاء) التي تتكون من مسافة واحدة أو أكثر (حتى ٢٠ حرف) أو أحرف إرجاع أو استجابة السطر أو علامات جدولة.

```
White Space
[3] S ::= (#x20 | #x9 | #xD | #xA)+
```

يتم تصنيف الأحرف للملائمة إلى أحرف أو أرقام أو أحرف أخرى. تتكون الأحرف من حرف أساسي أبجدي أو مقطعي ربما يكون متبوعاً بحرف مختلط أو أكثر أو بحرف رسومي. توجد تعريفات كاملة للأحرف المحددة في كل فئة في "ب" فئات الأحرف.

الاسم (**Name**): هو مقطع يبدأ بحرف أو أحد أحرف التنقيط ويستمر بمجموعة من الأحرف أو الأرقام أو الواصلات أو التسطير أسفل السطور أو علامات النقطتين أو النقاط والتي تسمى جميعاً بأحرف الاسم يتم الاحتفاظ بالأسماء التي تبدأ xml أو ما يمكن أن يطابقها (('X'|'x') ('M'|'m') ('L'|'l')) ليتم القياس عليها في هذه المواصفات أو في الإصدارات التالية لها.

ملحوظة (**Note**): حرف علامة النقطتين الموجود في أسماء XML يتم الاحتفاظ به للاختيار مع مسافات الأسماء من المتوقع أن تتم معايرة معناه مستقبلاً بحيث يمكن إن تحتاج المستندات التي تستخدم فيها علامة النقطتين لأغراض الاختبار إلى التحديث.(لا يوجد ما يضمن أن أي آلية لمسافة الاسم مستخدمة مع XML ستستخدم كمحدد لمسافة الاسم) عند التطبيق، يعني هذا أن الكاتبين يجب ألا يستخدموا علامة النقطتين في أسماء XML إلا في اختبارات مسافات الأسماء، لكن معالجات XML ستقبل علامة النقطتين كحرف اسم.

قطع الاسم (Nmtoken) هو أي خليط من أحرف الأسماء.

الأسماء والمقاطع

```
[4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':'
                 | CombiningChar | Extender
[5] Name      ::= (Letter | '_' | ':') (NameChar)*
[6] Names     ::= Name (S Name)*
[7] Nmtoken   ::= (NameChar)+
[8] Nmtokens  ::= Nmtoken (S Nmtoken)*
```

البيانات الحرفية هي أي سلسلة مقتبسة لا تحتوى على علامة الاقتباس المستخدمة كمحدد لتلك السلسلة. والقيم الحرفية تستخدم لتحديد محتوى الكائنات الداخلية (EntityValue) وقيم السمات (AttValue) والمعرفين الخارجيين (SystemLiteral). لاحظ أن SystemLiteral يمكن توزيعه دون البحث عن ترميز.

القيم الحرفية

```
[9]  EntityValue    ::= ' " ' ([^%&"] | PEReference | Reference)* ' " '
                        | " ' " ([^%&'] | PEReference | Reference)* " ' "
[10] AttValue       ::= ' " ' ([^<&"] | Reference)* ' " '
                        | " ' " ([^<&'] | Reference)* " ' "
[11] SystemLiteral  ::= (' " ' [^"]* ' " ') | (" ' " [^']* " ' ")
[12] PubidLiteral   ::= ' " ' PubidChar* ' " '
                        | " ' " (PubidChar – " ' ")* " ' "
[13] PubidChar      ::= #x20 | #xD | #xA | [a-zA-Z0-9]
                        | [-'()+,./:=?;!*#@$_%]
```

٢-٤ بيانات الأحرف والترميز

يتكون النص من بيانات حرفية وترميزات متداخلة. يأخذ الترميز شكل علامات البداية وعلامات النهاية وعلامات العناصر الفارغة ومراجع الكائنات ومراجع الأحرف والتعليقات ومحددات قسم CDATA وإعلانات أنواع المستندات وتعليمات المعالجة. وتكون جميع النصوص غير الخاصة بالترميز بيانات حرفية للمستند.

يمكن لحرف علامة الضم (&) وقوس الزاوية الأيسر (<) أن يظهرا في صورتها الحرفية فقط عند استعمالها لمحددي ترميز أو داخل تعليق أو تعليمات معالجة أو قسم CDATA ويكونان

صحيحين أيضاً في قيمة الكائن الحرفي لإعلان كائن داخلي. انظر "٤-٣-٢ الكائنات الموزعة جيدة التكوين". إذا كان من الضروري استخدامها في مكان آخر، فيجب تجاوزهما باستخدام مراجع أحرف رقمية أو بالسلاسل "&" و"<" على التوالي يمكن تمثيل قوس الزاوية الأيمن (>) باستخدام السلسلة">" ويجب (للتوافقية) أن يتم تجاوزه باستخدام > أو مرجع حرف عندما يظهر في السلسلة <[[في المحتوى، عندما لا تكون السلسلة في كفاية قسم CDATA.

في محتوى العناصر، البيانات الحرفية هي أي سلسلة من الأحرف لا تحتوي على محدد البداية لأي ترميز في قسم CDATA، البيانات الحرفية هي أي سلسلة من الأحرف لا تتضمن محدد نهاية قسم CDATA <[[.

للسماح لقيم السمات بأن تحتوي على كل من علامات الاقتباس المفردة والمزدوجة، يمكن تمثيل الفاصلة العليا أو علامة الاقتباس المفردة (')' وعلامة الاقتباس المزدوجة (")".

البيانات الحرفية

```
[14] CharData ::= [^<&]* - ([^<&]* ']]>' [^<&]*)
```

٢-٥ التعليقات

يمكن أن تظهر التعليقات (Comments) في أي مكان في مستند خارج الترميزات الأخرى. بالإضافة إلى ذلك، فقد تظهر داخل إعلان نوع المستند في أماكن مسموحة في القواعد، وهي ليست جزءاً من البيانات الحرفية للمستند يمكن لمعالج XML أن يجعل من الممكن للتطبيق أن يستخدم مع نصوص التعليقات، لكن ذلك ليس ضرورياً.للتوافق، يجب ألا توجد السلسلة (واصلتان متتاليتان) داخل التعليقات.

تعليقات

```
[15] Comment ::= '<!-' ((Char – '-') | ('-' (Char – '-')))* '->'
```

An example of a comment:

```
<!- declarations for <head> & <body> ->
```

٢–٦ تعليمات المعالجة

تسمح تعليمات المعالجة (PIs) للمستندات أن تحتوي على تعليمات للتطبيقات.

```
instructions applications
[16] PI ::= '<?' PITarget
            (S (Char* - (Char* '?>' Char*)))? '?>'
[17] PITarget  ::= Name - ((' X ' | ' x ') (' M ' | ' m ') (' L ' | ' l '))
```

ليست تعليمات المعالجة جزءاً من بيانات المستند الحرفية، لكنها يجب تمريرها إلى التطبيق. تبدأ تعليمات المعالجة بهدف (PITarget) يستخدم لتعريف التطبيق الذي ستم توجيه التعليمات له أسماء الأهداف XML و"xml" يتم الاحتفاظ بها ليتم معايرتها في هذه الإصدارة لهذه المواصفات أو الإصدارة التالية لها. يمكن استخدام آلية ملاحظات XML للإعلانات الرسمية لأهداف تعليمات البرمجة.

٢–٧ أقسام CDATA

يمكن أن توجد أقسام CDATA في أي مكان يمكن وجود بيانات حرفية فيه، وهي تستخدم لتجنب استخدام كتل النصوص التي تحتوي على أحرف يمكن أن يتم التعرف عليها كترميز تبدأ أقسام CDATA بالسلسلة]CDATA]!< وتنتهي بالسلسلة <[[.

أقسام CDATA

```
[18] CDSect  ::= CDStart CData CDEnd
[19] CDStart ::= '<![CDATA['
[20] CData   ::= (Char* - (Char* ']]>' Char*))
[21] CDEnd   ::= ']]>'
```

في قسم CDATA، يتم التعرف على السلسلة CDEnd فقط كترميز وبالتالي فإن أقواس الزوايا اليسرى وعلامات الضم يمكن أن توجد في صورتها الحرفية، فهي لا تحتاج (ولا يمكن) أن يتم تجنبها باستخدام < و& لا يمكن أن تتداخل أقسام CDATA وهذا مثال لقسم CDATA يتم التعرف فيه على <greeting>و</greeting> كبيانات حرفية وليست ترميزاً:

```
<![CDATA[<greeting>Hello, world!</greeting>]]>
```

٢-٨ Prolog وإعلان نوع المستندات

يمكن لمستندات XML أن تبدأ بإعلان XML الذي يحدد إصدارة XML المستخدمة "من المفضـــل وجودة" على سبيل المثال، فما يلي هو مستند XML كامل وجيد التكوين لكنه غير صحيح.

```
<?xml version="1.0"?> <greeting>Hello, world!</greeting>
```

وكذلك هذا المستند:

```
<greeting>Hello, world!</greeting>
```

يجب استـ ـرة 1.0 للإشارة إلى التوافق مع هذه الإصدارة لهذه المواصفـــات. فمن الخطأ استخدام القيمة 1.0 لمستند إذا لم يتطابق مع هذه الإصدارة لهذه المواصفات. تنـــوي مجموعة عمل NML إنتاج إصدارات أحدث لأرقام هذه المواصفات بعد 1.0 لكن ذلك لا يعنــــي أنهم يريدون إنتاج إصدارة مستقبلية XML أو استخدام خطة ترقيم معينة عند إنتاج أي منها. وبمـا أن الإصدارات المستقبلية غير مستبعدة، فإن هذا التكوين موجود للسماح بإمكانية التميز التلقـــائي للإصدارة إذا كان ذلك ضرورياً يمكن إن تحدد المعالجات خطأ إذا استقبلت مستندات ذات عناوين لإصدارات غير مدعمة لحقاً.

تكون وظيفة الترميز في مستند XML هي وصف بنيته التخزينية والمنطقية وإرفاق أزواج قيم السمات مع بنيتها المنطقية. توفر XML آلية هي إعلان نوع المستند لتعريف القيود علـــى البنيـــة المنطقية ولدعم استخدام وحدات التخزين المعرفة سابقاً. ويكون مستند XML صحيحـــاً إذا كـــان مرفقاً به إعلان نوع مستند وإذا كان المستند يحقق القيود الخاصة به يجب أن يظهر إعلان نـــوع المستند قبل العنصر الأول في المستند.

Prolog

```
[22] prolog          ::= XMLDecl? Misc* (doctypedecl Misc*)?
[23] XMLDecl         ::= '<?xml ' VersionInfo EncodingDecl? SDDecl? S?
'?>'
[24] VersionInfo     ::= S ' version ' Eq (' VersionNum '
                         | " VersionNum ")
[25] Eq              ::= S? '=' S?
[26] VersionNum      ::= ([a-zA-Z0-9_.:] | '-')+
[27] Misc            ::= Comment | PI |  S
```

يحتوي إعلان نوع مستند XML على أو يشير إلى إعلانات ترميزات تمثل قواعد لفئة من المستندات هذه القواعد معرفة باسم تعريف نوع المستند أو DTD. يمكن إن يشر إعلان نوع المستند إلى مجموعة جزئية (نوع خاص من الكائنات الخارجية) تحتوي على إعلانات ترميزات أو يمكن أن يحتوي على إعلانات الترميزات الموجودة مباشرة في مجموعة جزئية داخلية أو يمكن أن يحتوى على الاثنين. تعريف نوع المستند لأي مستند يتكون من كلا المجموعتين الفرعيتين مع بعضهما.

أما إعلان الترميز وهو إعلان عنصر أو إعلان قائمة سمات أو إعلان كائن أو إعلان ملاحظة يمكن أن تكون هذه الإعلانات مع بعضها البعض أو متفرقة في كائنان معاملات كما هو موضح في قيود جودة التكوين والصلاحية لاحقاً لمزيد من المعلومات، انظر ٤-الأبنية الحقيقية.

Document Type Definition

```
[28] doctypedecl ::= '<!DOCTYPE ' S Name (S ExternalID)?
                     S? ('[' (markupdecl | PEReference
                     | S)* ']' S?)? '>'
                     [ VC: Root Element Type ]
[29] markupdecl  ::= elementdecl | AttlistDecl
                     | EntityDecl | NotationDecl | PI
                     | Comment
                     [ VC: Proper Declaration/PE Nesting ]
                     [ WFC: PEs in Internal Subset ]
```

يمكن تكوين إعلانات الترميزات مع إبدال النص بكائنات المعاملات أو بدون ذلك تصف المنتجات لاحقاً في هذه المواصفات للنهايات غير الطرفية المفردة (إعلانات العناصر، إعلانات قائمة السمات، الخ) الإعلانات بعد تضمين جميع كائنات المعاملات.

قيد الصلاحية: نوع عنصر الجذر: يجب إن يطابق عنصر Name في تعريف نوع المستند نوع العنصر في عنصر الجذر.

قيد الصلاحية: التعريف الصحيح/ تضمين PE: يجب تضمين نص استبدال كيان المعامل بدقة باستخدام تعريفات الترميز وهذا يعني أنه إذا تم احتواء أول حرف أو آخر حرف لتعريف الترميز (إعلان الترميز فيما سبق) في النص الاستبدالى لمرجع كيان المعامل يجب احتواء كليهما في نفس النص الاستبدالي.

قيد جودة التكوين: PE في المجموعات الإضافية الداخلية: في المجموعات الإضافيــة التــي تخص DTD تقع مراجع كيانات المعاملات حيث تقع تعريفات الترميز وليــس داخــل تعريفــات الترميز ولا ينطبق هذا على المراجع التي تقع فـــي كيانـــات المعـــاملات الخارجيـــة أو علـــى المجموعات الإضافية الخارجية.

يجب أن تحتوي المجموعات الإضافية الخارجية وأي كيانات معاملات خارجية يشار إليها في DTD كما في المجموعات الإضافية الداخلية على سلسلة من تعريفات الترميز الكاملـــة لـــلأنواع التي تتيحها تعريفات الترميز للرموز غير الطرفية الموجود بها مسافات أو مراجع كينونة المعامل ومع ذلك يمكن تجاهل أجزاء من محتويات المجموعة الإضافية أو كينزنات المعـــامل الخارجيـــة بشروط عن طريق استخدام بناء الجزء الشرطي ولا يسمح بذلك فـــي المجموعـــات الإضافيـــة الداخلية.

المجموعة الإضافية الخارجية:

```
[30] extSubset ::=   TextDecl? extSubsetDecl
[31] extSubsetDecl ::=   ( markupdecl | conditionalSect |
```

تختلف المجموعات الإضافية الخارجية وكينونات المعامل الخارجي عن المجموعة الإضافيـــة الداخلية في أنه يسمح بوجود مراجع كينونة المعامل داخل.

تعريفات الترميز وليس فقط بين تعريفات الترميز.

```
<?xmlتversion="١,٠"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

مثال لمستند XML مع تعريف نوع المستند.

```
<?xml version=" "١,٠encoding="UTF-8" ?>
<!DOCTYPE greeting [
  <!ELEMENT greeting (#PCDATA)>
]>
```

يعطى hello.dtd وهو معرف النظام URI الذي يخص DTD المستند.

يمكن إعطاء التعريف محلياً كما يوضح المثال:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

The system identifier 'hello.dtd" gives the URI of a DTD for the document.

The declarations can also be given locally, as in this example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE greeting [
<!ELEMENT greeting (#PCDATA)>
]>
<greeting>Hello, world!</greeting>
```

في حالة استخدام المجموعات الإضافية الداخلية والخارجية يتوقع حدوث المجموعة الإضافية الداخلية قبل المجموعة الإضافية الخارجية ويكون تأثير ذلك هو أن تأخذ تعريفات قائمة السمة والكينونة في المجموعة الإضافة الداخلية الأسبقية عن المجموعة الإضافية الخارجية.

٩٠٢ تعريف المستند المنفرد

تؤثر تعريفات الترميز في محتوى المستند في أثناء تمريرها من معالج XML إلى تطبيق والأمثلة هي افتراضيات السمة وتعريفات الكينونة ويشير تعريف المستند المنفرد والذي يظهر كمكون لتعريف XML إلى ما إذا كان هناك مثل هذه التعريفات تظهر كأنها خارجة عن كينونة المستند.

تعريف المستند المنفرد

```
[32] SDDecl ::= S "standalone' Eq ((" ' " ('yes' | 'no')
            " ' ") | (' " ' ('yes' | 'no') ' " '))
            [ VC: Standalone Document Declaration ]
```

تشير القيمة yes في تعريف المستند المنفرد إلى عدم وجود تعريفات ترميز خارجة عن كينونة المستند سواء في المجموعة الإضافية الخارجية التي تخص DTD أو في كينونة معامل خارجي يرجع إليه من المجموعة الإضافية الداخلية مما يؤثر على المعلومات التي تمرر من معالج XML إلى التطبيق وتشير القيمة no إلى وجود أو إلى إمكانية وجود تعريفات ترميز خارجية ولا حظ أن تعريف المستند المنفرد يشير فقط إلى وجود تعريفات خارجية ولا يغير وجود مرجع للكينونات الخارجية في مستند من حالة الانفراد حتى ولو تم إعلان تلك الكينونات داخليا.

إذا لم يوجد تعريفات ترميز خارجية لا يكون لتعريف المستند المنفرد معنى وإذا وجد تعريفات ترميز خارجية ولم يوجد تعريف مستند منفرد يتم افتراض القيمة no.

يمكن لأي مستند XML يحتوى على no=standalone تحويله بأسلوب حسابي إلى مستند منفرد والذي يكون أفضل لتطبيقات توصيل شبكة الاتصال.

قيد الصلاحية: تعريف مستند منفرد: يجب أن تكون قيمة تعريف المستند المنفرد no في حالة احتواء أي تعريفات ترميز خارجية على أي من التعريفات التالية:

- السمات مع القيم الافتراضية، إذا ظهرت العناصر التي تطبق عليها تلك العناصر في المستند دون مواصفات قيم هذه السمات، أو.
- الكينونات (غير Amp,It gt ,opos,quot) إذا ظهرت تلك المراجع هنا الكينونات في المستند، أو.
- السمات مع القيم الخاضعة للتسوية حيث تظهر السمة في المستند مع قيمة تتغير نتيجة التسوية، أو
- أنواع العناصر مع محتوى العنصر، إذا ظهرت المسافة البيضاء مباشرة في أي مثال من هذه الأنواع.

ما يلي هو مثال تعريف XML في تعريف مستند منفرد:

```
<?xml version="1.0" standalone= 'yes'?>
```

٢-١٠ استخدام المسافات البيضاء

أثناء تحرير مستندات XML من المناسب استخدام المسافات البيضاء (مسافات، علامات تبويب والأسطر الفارغة والتي يرمز لها بحرف S غير طرفي هذه المواصفة) لإبعاد الترميز عن بعضه ليصبح مقروء بصورة أفضل وتلك المسافات البيضاء لا تضمن في الإصدار النهائي للمستند. ومن جهة أخرى فإن المسافات البيضاء الواجب حفظها في الإصدار النهائي معروفة وشائعة مثلاً في الشفرة تعليمات المصدر البرمجية.

يجب على معالج XML تمرير كل الأحرف في المستند التي لا تكون حروف ترميز إلى التطبيق كما معالج XML صالح التطبيق عن أي من هذه الحروف يكون المسافات البيضاء التي تظهر في محتوى العنصر.

قد يتم إرفاق سمة خاصة تسمى xml:space لعنصر لتشير إلى أن المسافات البيضاء في هذا العنصر يجب حفظها بواسطة التطبيقات. يجب في المستندات الصالحة أن يتم تعريف تلك السمة مثل أي شئ أخر في حالة استخدامها وعند تعريفها يجب إعطائها كنوع له ترقيم والتي تكون القيم المتاحة له هي "default" و"preserve" وعلى سبيل المثال:

```
<!ATTLIST poem    xml:space (default|preserve) 'preserve'>
```

تشير القيمة default إلى قبول العنصر لتطبيقات معالجة أنماط معالجة المسافات البيضاء أما القيمة preserve فإنها تشير إلى احتفاظ التطبيقات بكل المسافات البيضاء ويتم تطبيق هذا على كل العناصر داخل محتوى العنصر المحدد آلا في حالة التجاوز عنه بواسطة مثال آخر سمة .xml:space

لا يشير العنصر الجذري لأي مستند إلى شئ فيما يخص تطبيقات استخدام المسافات آلا في حالة تقديم قيمة لهذه السمة أو يتم تعريف السمة بقيمة افتراضية.

٢-١١ استخدام نهاية الخط

تخزين كينونات توزيع XML في ملف أجهزة الكمبيوتر والتي يتم تنظيمها لأغراض التحرير في أسطر وتفصل هذه الأسطر عن طريق دمج رموز أحرف الإرجاع (#xD) وتغذية الأسطر .(#xA)

لتبسيط مهام التطبيقات فأينما احتوت كينونة توزيع خارجية أو قيمة كينونة حرفية لكينونة توزيع داخلية على تتالي نهائي لحرفين #xD#xA أو حرف منفرد #xD يجب أن يمرر معالج XML للتطبيق الحرف الوحيد #xA ويمكن إنتاج هذا السلوك عن طريق تسوية كل فواصل الخطوط إلى #xA في أثناء الإدخال وقبل التوزيع.

٢-١٢ تعريف اللغة

من المفيد أثناء معالجة المستند تعريف اللغة الرسمية أو الطبيعة المكتوب بها المحتوى ويمكن إدراج سمة خاصة تسمى xml:lang في المستندات لتحديد اللغة المستخدمة في قيم السمة والمحتويات لأي عنصر في مستند XML في المستندات الصالحة مثل أي سمة آخرى يجب تعريف تلك السمة إذا استخدمت وتعتبر قيم هذه السمة هي معرفات اللغة كما تم توضيحها بواسطة:. [IETF RFC 1766], "Tags for the Identification of Languages"

تعريف اللغة

```
[33] LanguageID   ::= Langcode ('-' Subcode)*
[34] Langcode     ::= ISO639Code | IanaCode | UserCode
[35] ISO639Code   ::= ([a-z] | [A-Z]) ([a-z] | [A-Z])
[36] IanaCode     ::= ('i' | 'I') '-' ([a-z] | [A-Z])+
[37] UserCode     ::= ('x' | 'X') '-' ([a-z] | [A-Z])+
[38] Subcode      ::= ([a-z] | [A-Z])+
```

قد يكون Langcode أي مما يلي.

- تعليمات برمجة لغة تتكون من طرفين كما تم تعريفها بواسطة [ISO 639].
- معرف لغة مسجل مع Internet Assigned Numbers Authority [IANA] وتبدأ تلك المعرفات بحرف بادئ هو "-i"أو "-I"
- معرف لغة يعين المستخدم أو يتفق عليه عدة أطراف في مشروع خاص ويجب أن يبدأ المعرف بالحرف البادئ -x أو-X للتأكد من عدم التعارض مع أسماء قياسية أو مسجلة مع IANA.

قد يكون هناك أي عدد من أجزاء Subcode فإذا تواجد أول جزء من التعليمات البرمجية الإضافية وكان Subcode يحتوي على حرفين فلابد وأن يكون رمز الدولة من [ISO 3166] Codes for the representation of names of countries (رموز لتمثيل أسماء الدول). إذا كانت التعليمات البرمجية الإضافية تحتوي على أكثر من حرفين فيجب أن تكون تعليمات برمجية إضافية المستخدمة مسجلة بواسطة IANA إلا في حالة بدء Langcode بالحرف البادئ -x أو-X.

من المعتاد إعطاء رمز اللغة بحرف صغير ورمز الدولة إن وجد بحرف كبير لاحظ أن هذه القيم بخلاف الأسماء الأخرى في مستند XML تتأثر بحجم الأحرف.

فعلى سبيل المثال

```
<p xml:lang="en">The quick brown fox jumps over the lazy dog.</p>
p>
<p xml:lang="en-GB">What colour is it?</p>
<p xml:lang="en-US">What color is it?</p>
<sp who="Faust" desc='leise' xml:lang="de">
```

```
<l>Habe nun, ach! Philosophie,</l>
<l>Juristerei, und Medizin</l>
    <l>und leider auch Theologie</l>
    <l>durchaus studiert mit he i Bem Bem  h'n.</l>
    </sp>
```

البنية المعلن عنها مع xml:lang من المتوقع تطبيقها على كل السمات محتوى العنصر المحدد إلا في حالة التجاوز عنها مع مثال xml:lang على عنصر آخر داخل المحتوى نفسه.

تعريف بسيط xml:lang قد يأخذ الشكل التالي:

```
xml:lang  NMTOKEN  #IMPLIED
```

لكن قد يتم إعطاء قيم افتراضية محددة إذا كانت مناسبة. فعلى سبيل المثال مجموعة من الأشعار الفرنسية لطلاب يدرسون باللغة الإنجليزية حيث توجد الملحوظات على الأشعار باللغة الإنجليزية قد يتم تعريف سمة xml:lang كما يلي:

```
<!ATTLIST poem   xml:lang NMTOKEN 'fr'>
<!ATTLIST gloss  xml:lang NMTOKEN 'en'>
<!ATTLIST note   xml:lang NMTOKEN 'en'>
```

٣.الأبنية المنطقية

يحتوي كل مستند XML على عنصر أو أكثر وتعرف حدوده بعلامات بدء وعلامات نهاية أو بالنسبة للعناصر الفارغة بعلامة عنصر فارغ لكل عنصر نوع يعرف بالاسم وفي بعض الأحيان يسمى معرف عام (GI) اختصار لكلمة generic identifier وقد يكون له مجموعة من مواصفات السمات ولكل مواصفة سمة اسم وقيمة.

العنصر

```
[39] element ::= EmptyElemTag | STag content ETag
                 [ WFC: Element Type Match ]
                 [ VC: Element Valid ]
```

لا تقيد هذه المواصفة semantics، الاستخدام، أو (فيما ما وراء syntax) أسماء أنواع وسمات العنصر فيما عدا أن الأسماء التي تبدأ بمطابق (('X'|'x')('M'|'m')('L'|'l')) تحفظ لتكون قياسية في هذا الإصدار أو في إصدارات مستقبلية من هذه المواصفة.

قيد جودة –التكوين: مطابق نوع العنصر: يجب أن يطابق الاسم في علامة نهاية عنصر في علامة البدء.

قيد الصلاحية: صلاحية العنصر: يكون العنصر صالحاً إذا كان هناك تعريف يطابق elementdecl حيث يطابق الاسم نوع العنصر وتبقى أحد الأشياء التالية.

١- يطابق التعريف EMPTY ولا يكون للعنصر محتوى.

٢- يطابق التعريف children وتنتمي تتابع العناصر التابعة إلى اللغة الناتجة بواسطة التعبيرات العادية في نموذج المحتوي، مع مسافات بيضاء اختيارية (الأحرف التي تطابق حرف S غير الطرفي) بين كل زوج من العناصر التابعة.

٣- يطابق التعريف Mixed يتضمن المحتوى بيانات أحرف وعناصر تابعة التي تطابق أنواعها الأسماء في نموذج المحتوى.

٤- يطابق التعريف ANY ويتم تعريف أنواع أي عناصر تابعة.

٣-١ علامات البدء، علامات النهاية وعلامات العنصر الفارغة

يتم تعليم بداية كل عنصر XML غير فارغ بعلامة بدء علامة بدء start-tag.

Start-tag

```
[40] Stag       ::= '<' Name (S Attribute)* S? '>'
                [ WFC: Unique Att Spec ]
[41] Attribute  ::= Name Eq AttValue
                [ VC: Attribute Value Type ]
                [ WFC: No External Esssntity References ]
                [ WFC: No < in Attribute Values ]
```

يعطي Name في علامة البدء والنهاية type الخاص بالعنصر ويتم الإشارة إلى زوج Name-AttValue كمواصفات سمة attribute specifications العنصر، مع وجود Name في كل زوج يشار إليه كاسم سمة attribute name ومحتوي AttValue (النص بين محددات or كقيمة السمة attribute value.

قيد جودة التكوين Unique Att Spec (مواصفات سمة فريدة): لا يظهر اسم سمة أكثر من مرة واحدة في نفس علامة البدء علامة العنصر الفارغ.

قيد الصلاحية. نوع قيمة السمة: يجب إعلان السمة ويجب أن تكون السمة من النوع المعلـن لها (لمراجعة أنواع السمات راجع ٣,٣ إعلانات قائمة السمات).

قيد جودة التكوين: لا يوجد مراجع كينونات خارجية: لا يجب إن تحتوي قيم السمات علـــى مراجع كينونات مباشرة أو غير مباشرة لكينونات خارجية.

قيد جودة التكوين: لا توجد علامة، في قيم السمات: لا يجب أن يحتوي النص الاستبدالي لأي كينونة يشار إليها بصورة مباشرة أو غير مباشرة في قيمة السمة (بخلاف <) على علامة<

مثال لعلامة بدء:

```
<termdef id="dt-dog" term="dog">
```

يجب تعليم نهاية عنصر يبدأ بعلامة بدء بعلامة نهاية تحتوي على اسم يشير إلى نوع العنصر كما هو معطى في علامة البدء.

علامة النهاية End-tag

```
[42] ETag ::= '</' Name S? '>'
```

مثال لعلامة نهاية:

```
</termdef>
```

يسمى النص بين علامة البدء وعلامة النهاية محتوى العنصر.

محتوى العناصر

```
[43] content ::= (element | CharData | Reference |Sect | PI
                 | Comment)*
```

إذا كان عنصر empty يجب تمثيلها إما بواسطة علامة بدء يعقبها مباشرة علامة نهايـــة أو بواسطة علامة عنصر فارغ. يكون لعلامة العنصر الفارغ empty-element شكل خاص:

علامة للعناصر الفارغة

```
[44] EmptyElemTag ::= '<' Name (S Attribute)* S? '/>'
                      [ WFC: Unique Att Spec ]
```

يمكن استخدام علامات العنصر الفارغ لأي عنصر ليس له محتوى سواء تـــم تعريفـــه أو لا باستخدام الكلمة الأساسية EMPTY لإمكانية التشغيل المتداخل، يجب استخدام علامــــة العنصـــر الفارغ ويمكن استخدامها فقط للعناصر التي تم تعريفها EMPTY.

أمثلة للعناصر الفارغة:

```
<IMG align="left"
src="http://www.w3.org/Icons/WWW/w3c_home" />
<br></br>
<br/>
```

٣-٢ تعريفات أنواع العناصر

قد تقيد بنية عنصر مستند XML لأغراض الصلاحية باستخدام تعريفات قوائـــم الســـمات ونـــوع العنصر. تقيد تعريفات نوع العنصر محتوى العنصر.

تقيد تعريفات نوع العنصر أي أنواع العناصر يمكن ظهورها كتابع لعنصر يستطيع المســتخدم اختيار أن يقوم معالج XML بإصدار تحذير عندما يقوم التعريف بذكر نوع عنصر لا يتم تزويــد تعريف له ولكن هذا ليس خطأ.

يأخذ تعريف نوع العنصر الشكل التالي.

تعريف نوع العنصر

```
[45] elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'
                        [ VC: Unique Element Type Declaration ]
[46] contentspec ::= 'EMPTY' | 'ANY' | Mixed | children
```

حيث يعطي Name نوع العنصر الذي تم تعريفه.

قيد الصلاحية: تعريف نوع عنصر فريد: لا يجوز تعريف نوع عنصر أكثر من مرة واحدة.

أمثلة من تعريفات نوع العنصر:

```
<!ELEMENT br EMPTY>
<!ELEMENT p (#PCDATA|emph)* >
<!ELEMENT %name.para; %content.para; >
<!ELEMENT container ANY>
```

٣-٢-١ محتوى العنصر

نوع العنصر له محتوى عنصر في حين أن عناصر هذا النوع يجب أن تحتوي فقط على عناصر تابعة (لا بيانات أحرف) ويتم تفرقتها بصورة اختيارية عن طريق المسافات البيضاء (الأحـــرف التي تطابق S غير الطرفية) في تلك الحالة يتضمن القيد نموذج محتوي وصيغة برمجية بسـيطة تحكم على الأنواع المسموحة من العناصر التابعة والترتيب المسموح بظهورهم به تبنى الصيغـة البرمجية على أحرف المحتوى (cps) اختصار كلمة content particles التي تحتـوي علـى أسماء، قوائم اختيارات لأحرف المحتوى أو قوائم مرتبة لأحرف المحتوي.

نماذج محتوى العنصر

```
[47] children   ::= (choice | seq) ('?' | '*' | '+')?
[48] cp::= (Name | choice | seq) ('?' | '*' | '+')?
[49] choice     ::= '(' S? cp ( S? '|' S? cp )* S? ')'
                [ VC: Proper Group/PE Nesting ]
[50] seq        ::=  '(' S? cp ( S? ',' S? cp )* S? ')'
                [ VC: Proper Group/PE Nesting ]
```

حيث أن كل Name هو نوع العنصر الذي قد يظهر كتابع أي حرف محتوي في قائمة اختيار قد يظهر في محتوى العنصر في موقع ظهور قائمة الاختيار في الصيغة البرمجيـة. يجب أن تظهر كل أحرف المحتوى التي تقع في قائمة ترتيب في محتوى العنصر بالترتيب المعطي فـي القائمة.

الأحرف الاختيارية التي تلي اسم أو قائمة تحكم إذا ما كان العنصر أو أحرف المحتوى فـي القائمة قد يظهروا مرة أو أكثر (+)، صفر أو أكثر (*) أو صفر أو مرة مضاعفه (?). يعنـي اختفاء مثل هذا العامل أن العنصر أو أحرف المحتوى يجب أن يظهر مرة واحـدة فقـط هـذا syntax والمعني متشابهين تماماً مع المستخدمين في إنتاج هذه المواصفة.

يطابق محتوى العنصر نموذج محتوى في حالة إمكانية تعقب مسار خلال نموذج المحتـوى، متبعاً الترتيب والاختيار وتكرار العوامل ومطابقة كل عنصر في المحتوى مع نوع عنصر فـي نموذج المحتوى. للتوافق فإن من الخطأ إذا أمكن لعنصر في المستند يطابق أكثر من حدوث واحد لنوع عنصر في نموذج المحتوى للحصول على مزيداً من المعلومات راجع الفقرة هـ أنـواع المحتويات المحددة.

قيد الصلاحية المجموعة المناسبة Proper Group/PE Nesting: يجب وضع النص الاستبدالى لكينونة المعامل داخل مجموعات أقواس هذا يعني أنه إذا تم احتواء أي من أقواس البداية والنهاية في بنية choice, seq, Mixed في النص الاستبدالي لكينونة معامل يجب تضمين كلاهما في نفس النص الاستبدالى لإمكانية التشغيل المتداخل إذا ظهر مرجع كينونة معامل في بنية choice, seq, or Mixed لا يجب أن يكون النص الاستبدالي فارغ ولا يجب أن يكون أول أو آخر حرف غير فارغ للنص الاستبدالي من الأدوات الرابطة (و أو ١).

أمثلة لنماذج محتوي العنصر

```
<!ELEMENT spec (front, body, back?)>
<!ELEMENT div1 (head, (p | list | note)*, div2*)>
<!ELEMENTتdictionary-body (%div.mix; | %dict.mix;)*>
```

٣-٢-٢ المحتوى المختلط

يكون لنوع العنصر محتوى مختلط عندما تحتوي عناصر من هذا النوع على بيانات حرفية، واختيارياً يتم تداخل تلك البيانات مع العناصر التابعة في هذه الحالة، قد يتم تقييد العناصر التابعة لهذه الأنواع ولكن ليس ترتيبهم أو عدد مرات حدوثهم.

تعريفات المحتوى المختلط

```
[51] Mixed ::= '(  S? '#PCDATA' (S? '|' S? Name)* S? ')* '
             | ;(' S? '#PCDATA' S? ')'
             [ VC: Proper Group/PE Nesting ]
             [ VC: No Duplicate Types ]
```

تعطي Names أنواع العناصر التي قد تظهر كعناصر تابعة.

قيد الصلاحية:لا أنواع مزدوجة: لا يجب أن يظهر الاسم أكثر من مرة في تعريف محتوى مختلط منفرد.

أمثلة من تعريفات المحتوى المختلط:

```
<!ELEMENT p (#PCDATA|a|ul|b|i|em)*>
<!ELEMENT p (#PCDATA | %font; | %phrase; | %special; |
%form;)* >
```

```
<!ELEMENT b (#PCDATA)>
```

3.3 Attribute-List Declarations

٣-٣ تعريفات قوائم السمات

تستخدم السمات لربط زوج قيمة الاسم مع العناصر قد تظهر مواصفات السمات فقط داخل علامات البدء وعلامات العنصر الفارغ ولذا فإن الإنتاج المستخدم لإعادة ترتيبهم يظهر في ١,٣ علامات البدء علامات النهاية وعلامات العنصر الفارغ يمكن استخدام تعريفات قوائم السمات.

- لتعريف مجموعة السمات المتصلة بنوع عنصر معطي.
- لوضع قيود الأنواع لهذه السمات.
- لتقديم القيم الافتراضية لهذه السمات\

تعريفات قوائم السمات: تحدد اسم ونوع بيانات والقيمة الافتراضية (إن وجدت) لكل سمة مرتبطة مع نوع عنصر معطي.

تعريفات قوائم السمات

```
[52] AttlistDecl ::= '<!ATTLIST' S Name AttDef* S? '>'
[53] AttDef      ::= S Name S AttType S DefaultDecl
```

الاسم Name في قاعدة AttlistDecl هو نوع عنصر وباختيار المستخدم يمكن أن يصدر معالج XML تحذير إذا تم إعلان السمات لنوع عنصر لم يتم تعريفه هو ولكن لا يعتبر ذلك خطأ الاسم Name في قاعدة AttDef هو اسم السمة.

عند تقديم أكثر من AttlistDecl لنوع عنصر معطي يتم دمج محتويات كل أنواع العناصر المعطاة عند تقديم أكثر من تعريف واحد لنفس السمة في نوع عنصر معطي يكون التعريف الأول هو التعريف الملزم ويتم تجاهل التعريف الأخير. لإمكانية التشغيل المتداخل، قد يختار DTDs تقديم تعريف قائمة سمة واحد على الأكثر لنوع عنصر معطي وعلى الأكثر تعريف سمة واحد لاسم سمة معطي وعلى الأقل تعريف سمة واحد لكل تعريف قوائم سمة لإمكانية التشغيل المتداخل؟ يمكن لمعالج XML عند اختيار المستخدم إصدار تحذير عند تقديم أكثر من تعريف قائمة سمة واحد لنوع عنصر معطي أو عند تقدم أكثر من تعريف سمة واحد لسمة معطاة ولكن يعتبر هذا خطأ.

٣-٣-١ أنواع السمات

أنواع سمات XML هي ثلاثة أنواع هم نوع سلسلة ومجموعة أنواع رمزية وأنواع عددية يـــأخذ نوع السلسلة أي سلسلة حرفية كقيمة أما بالنسبة لمجموعة الأنواع الرمزية فلــها قيــود lexical وsemantic متغيرة كما يوضح ما يلي.

أنواع السمات

```
[54] AttType ::= StringType | TokenizedType | EnumeratedType
[55] StringType ::= 'CDATA'
[56] TokenizedType ::= 'ID'        [ VC: ID ]
                                   [ VC: One ID per Element Type ]
                                   [ VC: ID Attribute Default ]
                    | 'IDREF'      [ VC: IDREF ]
                    | 'IDREFS'     [ VC: IDREF ]
                    | 'ENTITY'     [ VC: Entity Name ]
                    | 'ENTITIES'   [ VC: Entity Name ]
                    | 'NMTOKEN'    [ VC: Name Token ]
                    | 'NMTOKENS'   [ VC: Name Token ]
```

قيد الصلاحية: معرف ID: يجب أن تطابق قيم نوع المعرف ID إنتاج Name لا يجـــب أن يظهر الاسم أكثر من مرة واحدة في مستند XML كقيمة لهذا النوع مثل: يجب أن تعرف قيــم ID العناصر التي تحملها.

قيد الصلاحية: معرف واحد لكل نوع عنصر: لا يجب إن يكون لنوع عنصر أكثر من ســـمة معرف واحدة محددة.

قيد الصلاحية: معرف السمة الافتراضي: يجب أن يكون لسمة المعرف افتراضي معرف لكل من IMPLIED# أو REQUIRED#.

قيد الصلاحية IDREF: يجب أن تطابق قيم نوع IDREF إنتاج Name ويجب أن تطابق قيم نوع IDREFS الأسماء Names، ويطابق كل اسم قيمة سمة معرف ID على عنصر في مســتند XML فمثلا يجب أن تطابق قيم IDREF قيمة بعض سمات المعرف ID.

قيد الصلاحية: اسم الكينونة: يجب أن تطابق القيم من نــوع Name ENTITY، ويجــب إن يطابق كل Name اسم كينونة غير موزعة معرفة في DTD.

قيد الصلاحية: رمز الاسم: يجب أن تطابق قيم نوع NMTOKEN إنتاج Nmtoken وتطــابق قيم نوع Nmtokens NMTOKENS.

السمات العددية: يمكن أن تأخذ واحدة من قائمة القيم المعطاة في التعريف هناك نوعان مــن الأنواع العددية.

أنواع السمات العددية

```
[57] EnumeratedType ::= NotationType | Enumeration    ss
[58] NotationType ::= 'NOTATION' S '(' S? [ VC: Notation Attributes]
                Name (S? '|' S? Name)* S? ')'
[59] Enumeration  ::= '(' S? Nmtoken (S?  [ VC: Enumeration ]
                '|' S?Nmtoken)* S? ')'
```

تعريف سمة NOTATION مجموعة رموز معرفة في DTD مع أنظمة متصلـــة بــها و/أو معرفات عامة ليتم استخدامها في ترجمة العنصر الذي يتم إلحاق السمة به.

قيد الصلاحية: سمات مجموعة الرموز: يجب أن تطابق قيم هذا النوع أحد أسماء مجموعـات الرموز المضمنة في التعريف ويجب تعريف كل أسماء مجموعـــات الرمـوز المضمنــة فــي التعريف.

قيد الصلاحية: العدد يجب أن تطابق قيم هذا النوع أحد رموز Nmtoken أكثر مـــن مــرة واحدة في أنواع السمات العددية لنوع عنصر منفرد.

لإمكانية التشغيل المتداخل، لا يجب أن يحدث نفس Nmtoken أكثر من مرة واحدة في أنواع السمات العددية لنوع عنصر منفرد.

٣-٣-٢ افتراضيات السمة

تقدم تعريف السمة معلومات عن ما إذا كان سيطلب وجود السّمة وإذا لم يطلب وجودها فيعطــي معلومات عن كيفية تفاعل معالج XML إذا اختفت سمة معرفة في مستند.

افتراضيات السمة

```
[60] DefaultDecl ::= '#REQUIRED' | '#IMPLIED'
                   | (('#FIXED' S)? AttValue)
                     [ VC: Required Attribute ]
                     [ VC: Attribute Default Legal ]
                     [ WFC: No < in Attribute Values ]
                     [ VC: Fixed Attribute Default ]
```

في تعريف سمة، يعني #REQUIRED أنه يجب تقديم السـمة بصـورة مسـتمرة وتعني #IMPLIED أنه لا يتم تقديم أي قيمة افتراضيـة. إذا كـان التعريـف ليـس #REQUIRED أو#IMPLIED تحتوي قيمة AttValue على قيمة default المعرفة وتعلـن الكلمـة الأساسـية #FIXED أن السمة يجب أن تحصل على القيمة الافتراضية بصفة مستمرة. إذا تم تعريف قيمـة افتراضية عندما يواجه معالج XML سمة محذوفة فإنه يتصرف كما لو كانت السمة موجودة مـع القيمة الافتراضية المعرفة.

قيد صلاحية السمة المطلوبة: إذا كـان التعريـف الافـتراضي هـو الكلمـة الأساسـية: #REQUIRED فإن السمة يجب تحديدها لكل العناصر من هذا النوع في تعريف قائمة السمة.

قيد الصلاحية Legal افتراض السمة: يجب أن تلبي قيمة الافتراض المعرفة قيـود Lexicel لنوع السمة المعرف.

قيد الصلاحية: Legal افتراض السمة الثابت: إذا كان لسمة قيمة افتراضية معرفة مع الكلمة الأساسية #FIXED يجب أن تطابق أمثلة هذه القيمة الافتراضية.

أمثلة تعريفات قوائم السمات:

```
<!ATTLIST termdef
    id     ID    #REQUIRED
    name   CDATA  #IMPLIED>
<!ATTLIST list
     type   (bullets|ordered|glossary)  "ordered">
<!ATTLIST form
       method  CDATA   #FIXED "POST">
```

٣-٣-٣ تسوية قيم السمات

قبل تمرير قيمة سمة تطبيق أو يتم التحقق منها للتأكد من صلاحيتها يجب أن يقوم معـــالج XML بتسويتها كما يلي.

- يتم معالجة مرجع الحرف بواسطة إلحاق الحرف المرجع إلى قيمة السمة.
- يتم معالجة مرجع كينونة عن طريق معالجة النص الاستبدالي للكينونة بصوره متكررة.
- تتم معالجة حرف مسافة بيضاء (#x20, #xD, #xA, #x9) عن طريق إلحاق #x20 إلى القيمة المسواة إلا أنه يتم إلحاق #x20 فردية لترتيب #xD#xA، الذي يكون جـزء من كينونة توزيع خارجية أو قيمة كينونة حرفية لكينونة توزيع داخلية.
- يتم معالجة أحرف بواسطة إلحاقهم إلى القيمة المسواة

إذا لم تكن القيمة المعرفة CDATA يجب أن يعالج معالج XML قيمة السمة المسواة عن طريق حذف أي مسافات أحرف بادئة أو نهائية (#x20) وباستبدال ترتيبات أحرف المسافات (#x20) بمسافة حرف فردية (#x20).

كل السمات التي يقرأ لها تعريف. يجب أن تعامل بموزع غير صالح كما لو كـــانت أعلنـت CDATA.

٣-٤ المقاطع الشرطية

تعتبر المقاطع الشرطية أجزاء من المجموعة الإضافية الخارجية لتعريف نوع المستند التي يتـــم تضمينها في، أو استبعادها من، البنية المنطقية التي تخص DTD التي تعتمد على كلمات أساســية تحكمهم.

المقاطع الشرطية:

```
[61] conditionalSect    ::=  includeSect | ignoreSect
[62] includeSect   ::= '<![' S? 'INCLUDE' S? '[' extSubsetDecl ']]>'
[63] ignoreSect    ::= '<![' S? 'IGNORE' S? '[' ignoreSectContents*
                    ']]>'
[64] ignoreSectContents ::= Ignore ('<![' ignoreSectContents
                         ']]>' Ignore)*
[65] Ignore             ::= Char* - (Char* ('<![' | ']]>') Char*)
```

مثل مجموعات DTD الإضافية الخارجية والداخلية قد يحتوى مقطع شرطي على واحد أو أكثر من التعريفات الكاملة، التعليقات والتعليمات البرمجية أو مقاطع شرطية مضمنة مختلطة مع مسافات بيضاء.

إذا كانت الكلمة الأساسية للمقطع الشرطي هي INCLUDE. إذاً فمحتويات المقطع الشرطي هي جزء من DTD إذا كانت الكلمة الأساسية للمقطع الشرطي هي IGNORE فمحتويات المقطع الشرطي ليست جزء منطقي من DTD لا حظ أنه للحصول على توزيع يعتمد عليه فمحتويات حتى المقاطع الشرطية التي تم تجاهلها يجب قراءتها من أجل تحديد المقاطع الشرطية المضمنة وتأكيد أن نهاية المقاطع الشرطية (المتجاهلة) الخارجية يتم تحديدها بصورة صحيحة إذا وقع مقطع شرطي بكلمة أساسية هي INCLUDE داخل مقطع شرطي أكبر بكلمة أساسية هي IGNORE يتم تجاهل كلاً من المقاطع الشرطية الداخلية والخارجية.

إذا كانت الكلمة الأساسية للمقطع الشرطي هي مرجع كينونة معامل، يجب استبدال كينونة المعامل بواسطة محتوياته قبل أن يقرر المعالج تضمين أو تجاهل المقطع الشرطي.

مثال:

```
<!ENTITY % draft 'INCLUDE' >
<!ENTITY % final 'IGNORE' >

<![%draft;[ <!ELEMENT book (comments*, title, body,
supplements?)>
]]>
<![%final;[
<!ELEMENT book (title, body, supplements?)>
]]>
```

٤. الأبنية الحقيقية

قد يحتوي مستند XML على وحدة أو أكثر من وحدات التخزين. تلك الوحدات تسمى كينونات entities وكلهم لهم محتوى content ويتم تعريفهم كلهم (ماعدا كينونة المستند، راجع ما يلي ذلك مجموعة DTD الإضافية الخارجية) بواسطة name (اسم) لكل مستند XML كينونة تسمي كينونة المستند والتي تؤدي دور نقطة البدء لمعالج XML وقد تحتوي على المستند بأكمله.

يمكن توزيع أو عدم توزيع الكينونات. يتم الإشارة إلى محتويات كينونة موزعة علـــى إنـــها النص الاستبدالي ويعتبر هذا النص جزء مكمل للمستند.

تعتبر الكينونة غير الموزعة مصدر قد تكون محتوياته نص أو قد لا تكون نص وإذا كـــانت نص قد لا تكون XML لكل كينونة غير موزعة مجموعة رموز متصلة بها ومعرفة بالاسم. بعـــد المطالب ة بأن يجعل معالج XML المعرفات للكينونة ومجموعة الرموز متاحة لتطبيق لا يضــــع XML قيود على المحتويات للكينونات غير الموزعة.

Parsed entities are invoked by name using entity references; unparsed entities by name, given in the value of ENTITY or ENTITIES attributes.

الكينونات العامة: هي كينونات للاستخدام داخل محتوي المستند. في هـــــذه المواصفـــة يتـــم الإشارة إلى الكينونات العامة بالمصطلح غير المؤهل entity (كينونة) عندما لا يؤدي ذلك إلـــى غموض تعتبر كينونات المعامل كينونات موزعة للاستخدام في DTD يستخدم هذين النوعين مـــن الكينونات نماذج استرجاع مختلفة ويتم تعريفهما في سياقات مختلفة. بالإضافة إلى ذلك فهم يقعـوا في حيز أسماء مختلف؟ كينونة معامل وكينونة عامة بنفس الاسم هي كينونتين مختلفتين.

٤-١ مراجع الأحرف والكينونة

يشير مرجع الحرف إلى حرف محدد في مجموعة أحرف ISO/IEC 10646، فعلى سبيل المثال واحدة لا يتم الوصول إليها بصورة مباشرة من أجهزة الإدخال المتاحة.

مرجع الحرف

```
[66] CharRef ::= '&#'[0-9]+ ';'
               | '&#x'[0-9] a-fA-F]+ ';' [ WFC: Legal Character ]
```

قيد جودة التكوين: الأحرف Legal: الأحرف المشار إليها باستخدام مراجع الحرف يجــــب مطابقتها مع إنتاج Char.

إذا بدأ مرجع الحرف بعلامة &#x، الأرقام والحروف إلى النهاية وتقديم تسهيل ست عشــري لنقطة رمز الحرف في ISO/IEC 10646 إذا بدأت فقط بعلامة #& تقدم الأرقام إلـــى النـــهاية تمثيل عشري لنقطه رمز الحرف.

يشير مرجع الكينونة إلى محتوى الكينونة المسماة تستخدم المراجع للكينونات العامة الموزعة علامة ضم (&) وفاصلة منقوطة (;) كمحددة تستخدم مراجع كينونة المعامل علامة النسبية (%) والفاصلة المنقوطة (;) كمحددات.

مرجع الكينونة

```
[67] Reference      ::= EntityRef | CharRef
[68] EntityRef      ::= '&' Name ';'  [ WFC: Entity Declared ]
                                      [ VC: Entity Declared ]
                                      [ WFC: Parsed Entity ]
                                      [ WFC: No Recursion ]
[69] PEReference  ::= '%' Name ';' [ VC: Entity Declared ]
                                      [ WFC: No Recursion ]
                                      [ WFC: In DTD ]
```

قيد جودة التكوين: تعريف الكينونة في مستند دون أي DTD، مستند له مجموعة DTD إضافية داخلية لا تحتوي على مراجع كينونة معامل أو مستند يكون standalone=yes يجب أن يطابق Name المعطي في مرجع الكينونة في تعريف كينونة، فيما عدا المستندات جيدة التكوين لا تحتاج إلى تعريف أي من الكينونات التالية: lt, gt, apos, quot يجب أن يسبق تعريف كينونة معامل أي مرجع له وبالمثل تعريف كينونة عامة يجب أن يسبق أي مرجع له يظهر في قيمة افتراضية في تعريف سمة. لاحظ أنه إذا تم تعريف كينونات في المجموعة الإضافية الخارجية أو في كينونات المعامل الخارجية ولا يلزم معالج عدم الصلاحية قراءة ومعالجة تلك التعريفات ولمثل هذه المستندات، القاعدة أن كينونة يجب أن تعرف من قيود جودة التكوين فقط في حالة أن تكون standalone=yes.

قيد الصلاحية: تعريف الكينونة في مستند له مجموعة إضافية خارجية أو كينونات معامل خارجية بها standalone=no يجب أن يطابق Name المعطى في مرجع الكينونة الموجود في تعريف الكينونة. لإمكانية التشغيل المتداخل، يجب أن تعلن المستندات الصالحة عن كينونات amp, lt, gt, apos, quot في النموذج المحدد في ٦,٤ الكينونات المعرفة سابقاً يجب أن يسبق تعريف كينونة عامة أي مرجع له يظهر في قيمة افتراضية في تعريف قائمة السمة.

قيد جودة التكوين: الكيونات الموزعة: لا يجب أن يحتوى مرجع كينونة على اسم كينونة غير موزعة. يمكن الإشارة إلى الكينونات غير الموزعة فقط في قيم السمات المعرفة على أنها

من نوع ENTITY أو ENTITIES. قيد جودة التكوين: No Recursion: لا يجب أن تحتـــوى الكينونة الموزعة على مرجع متكرر لها سواء بصورة مباشرة أو غير مباشرة.

قيد جودة التكوين: في DTD تظهر مراجع كينونة المعامل في DTD تظهر مراجع كينونـــة المعامل في DTD فقط.

أمثلة لمراجع الكينونة والحرف:

```
Type <key>less-than</key> (&#x3C;) to save options.
This document was prepared on &docdate; and
Is classified &security-level;.
```

أمثلة لمرجع كينونة معامل:

```
<!- declare the parameter entity "ISOLat2"... ->
<!ENTITY % ISOLat2
        SYSTEM "http://www.xml.com/iso/isolat2-xml.entities" >
<!-    now reference it. ->
%ISOLat2;
```

٤–٢ تعريفات الكينونات

تعريف الكينونات مثل ما يلى.

تعريف الكينونة

```
[70] EntityDecl     ::= GEDecl | PEDecl
[71] GEDecl         ::= '<!ENTITY' S Name S EntityDef S? '>'
[72] PEDecl::= '<!ENTITY' S '%' S Name S PEDef S? '>'
[73] EntityDef      ::= EntityValue | (ExternalID NDataDecl?)
[74] PEDef          ::= EntityValue | ExternalID
```

يعرف Name الكينونة في مرجع الكينونة أو في حالة الكينونة غير الموزعة يعـرف Name الكينونة في سمة ENTITY أوENTITIES إذا تم تعريف نفس الكينونة أكثر من مرة يكـــون أو تعريف يقابل المستخدم هو التعريف الملزم. وباختيار المستخدم يمكن أن يصـــدر معـــالج XML تحذير تعريف الكينونات عدة مرات.

٤-٢-١ الكينونات الداخلية

إذا كان تعريف الكيان هو EntityValue، تسمي الكينونة المعرفة كينونة داخلية internal entity لا يوجد كائن تخزين حقيقي منفصل ويتم إعطاء محتوي الكينونة في التعريف. لا حظ أنه قد يطلب بعض المعالجة لمراجع الكينونة والحرف في قيمة الكينونة الحرفية وذلك لإنتاج النص الاستندالي الصحيح راجع ٥,٤ بناء نص استبدال الكائنات الداخلية.

الكينونة الداخلية هي كينونة موزعة مثال لتعريف كينونة داخلية.

```
<!ENTITY Pub-Status "This is a pre-release of the
specification ">
```

٤-٢-٢ الكينونات الخارجية

إذا لم تكن الكينونة داخلية فهي كينونة خارجية external entity تعريف كما يلي

تعريف كينونة خارجية

```
[75] ExternalID      ::= 'SYSTEM' S SystemLiteral
                        | 'PUBLIC' S PubidLiteral S SystemLiteral
[76] NdataDecl       ::= S 'NDATA' S Name [ VC: Notation
Declared ]
```

إذا كانت NdataDecl ظاهرة فهذه كينونة غير موزعة عامة أما في أي حالة أخرى فهي كينونة موزعة.

قيد الصلاحية: تعريف مجموعة الرموز: يجب أن يطابق name الاسم المعرف لمجموعة الرموز.

تسمى SystemLiteral معرف نظام الكينونة وهي عبارة عن URI قد تستخدم لاسترجاع الكينونة لا حظ أن علامة (#) ومعرف الجزء المستخدم بصورة متكررة مع URIs ليس بصفة رسمية، جزء من URI نفسها. قد يشير معالج XML إلى خطأ إذا تم إعطاء معرف جزء كجزء من معرف النظام. إلا إذا تم تزويدها بمعلومات من خارج نطاق هذه المواصفة (مثل نوع عنصر XML خاص معرف بواسطة DTD محدد أو تعليمات معالجة معرفة بواسطة مواصفة تطبيق خاصية) فإن URI النسبية تكون منسوبة إلى موقع المصدر الذي يقع داخله تعريف الكينونة وبذلك يكون URI ذو صلة بكينونة المستند أو بالكينونة التي تحتوي على مجموعة DTD الاضافية الخارجية أو بكينونة معامل خارجية أخرى.

يجب أن يستخدم معالج XML حرف لا يتيح ASCII في URI عن طريق تمثيل الحرف في UTF& كبايت أو أكثر ثم بعد ذلك الهروب من وحدات البايت تلك باستخدام آلية هروب URI (أي عن طريق تحويل كل بايت إلى HH% حيث تعني HH مجموعة رموز الست عشري لقيمة البايت).

بالإضافة إلى معرف النظام، قد يتضمن معرف خارجي على معرف خارجي على معرف عام pudlic identifier. معالج XML الذي يحاول استرجاع محتوى الكينونة قد يستخدم المعرف العام لمحاولة إنتاج URI بديل. إذا لم يستطيع المعالج تنفيذ ذلك يجب أن يستخدم URI المحددة في النظام حرفياً قبل إجراء مطابقة يجب تسوية كل سلاسل المسافات البيضاء في المعرف العام إلى حروف مسافات (#x20) ويجب حذف المسافات البيضاء البادئة والخاتمة.

أمثلة من تعريفات كينونة خارجية:

```
<!ENTITY open-hatch SYSTEM
    "http://www.textuality.com/boilerplate/OpenHatch.xml">
<!ENTITY open-hatch
    PUBLIC "-//Textuality//TEXT Standard open-hatch boilerplate//EN"
    "http://www.textuality.com/boilerplate/OpenHatch.xml">
<!ENTITY hatch-pic SYSTEM "../grafix/OpenHatch.gif" NDATA gif >
```

٤-٣ الكينونات الموزعة

٤-٣-١ تعريفات النص

قد نبدأ الكينونات الموزعة الخارجية بتعريف نص text declaration.

تعريف نص

```
[77] TextDecl ::= '<?xml' VersionInfo? EncodingDecl S? '?>'
```

يجب تقديم تعريف النص حرفياً، وليس بالإشارة إلى كينونة موزعة. لا يظهر أي تعريف نص في أي موقع إلا في بداية كينونة موزعة خارجية.

٤-٣-٢ الكائنات الموزعة جيدة التكوين

تكون كينونة المستند جيده التكوين إذا كانت تطابق مستند تسمية الإنتاج. تكون كينونة موزعة عامة خارجية جيدة التكوين إذا كانت تطابق الإنتاج المسمى extParsedEnt.تكون كينونة المعامل الخارجية جيدة التكوين إذا كانت تطابق الإنتاج المسمى EYTPE.

كينونة موزعة خارجية جيدة التكوين

```
[78] extParsedEnt     ::= TextDecl? content
[79] extPE            ::= TextDecl? extSubsetDecl
```

تكون الكينونة الموزعة العامة الداخلية جيدة التكوين إذا تطابق نصها الاستبدالي مع محتوى تسمية الإنتاج تكون كينونات المعامل الداخلية جيدة التكوين بواسطة التعريف.

نتيجة جودة التكوين في الكينونات هو أن البنيات الحقيقية والمنطقية في مستند XML يتم تضمينها بصورة صحيحة. لا يمكن أن تبدأ علامة بدء أو علامة نهاية أو علامة عنصر فارغ أو عنصر أو تعليق أو تعليمات معالجة أو مرجع حرف أو مرجع كينونة في كينونة وينتهي في آخري.

٤-٣-٣ تشفير الأحرف في الكينونات

قد يستخدم كل كينونة موزعة خارجية في مستند XML قادرة على قراءة الكينونات أما في UTF-8 أو UTF-16.

يجب أن تبدأ الكينونات المشفرة في UTF-16 مع Byte Order Mark المشروحة بواسطة ISO/IEC 10646 Annex E و unicode ملحق "ب" (حرف ZERO WIDTH NO-BREAK SPACE،#xFEFF) يكون هنا عبارة عن إمضاء مشفرة ليس جزء من الترميز أو بيانات الحرف لمستند XML يجب أن يستطيع معالج XML استخدام هذا الحرف للتفرقة بين المستندات المشفرة UTF-8 أو UTF-16.

على الرغم من أنه مطلوب من معالج XML قراءة الكينونات فقط في تشفير UTF-8 و UTF-16، ومن المعروف أن شفرات أخرى تستخدم حول العالم وقد يكون من الأفضل لمعالجي XML قراءة الكينونات التي تستخدمهم. الكينونات الموزعة التي يتم تخزينها في شفرة غير شفرات UTF-8 أو UTF-16 يجب أن تبدأ بتعريف نص يحتوي على تعريف شفرة.

تعريف الشفرة

```
[80] EncodingDecl ::= S 'encoding' Eq ('"' EncName '"'
                       | "'" EncName "'" )
[81] EncName      ::= [A-Za-z] ([A-Za-z0-9._] | '-')*
/* Encoding name contains only Latin characters */
```

في كينونة المستند، يكون تعريف التشفير جزء من تعريف XML ويكون EncName هو اسم التشفير المستخدم.ISO-8859-1 وISO-8859-2 وISO-8859-9 ... أجزاء

في تعريف شفرة، يجب استخدام قيم UTF-8, UTF-16 و ISO-10646-UCS-2 ISO-10646-UCS-4 للشفرات المتعددة وتحويلات Unicode / ISO/IEC 10646. وتستخدم قيم ISO 8859 وتستخدم قيم ISO-2022-JP وShift_JIS وEUC-JP لنماذج الشفرات المتعددة من IS X-0208-1997 تستطيع معالجات XML التعرف على شفرات الأحرف المسجلة (مثل charsets) مع Internet Assigned Numbers Authority [IANA] غير المدرجين في القائمة يتم الإشارة إليهم باستخدام أسمائهم المسجلة لاحظ أن الأسماء المسجلة معرفة على أنها تتبع حالة الحرف سواء حرف كبير أو صغير ولذا فالمعالجات التي ترغب في مطابقتهم يجب اتباع نفس الأسلوب في حالة الأحرف.

في غياب المعلومات المقدمة بواسطة بروتوكول نقل خارجي مثل HTTP أوMIME فمن الخطأ لكينونة تتضمن تعريف تشفير أن يتم تقديمها إلى معالج XML في تشفير غير المسمى في التعريف ليقع تعريف شفرة في مكان غير بداية الكينونة الخارجية أو لكينونة تبدأ أما مع Byte Order Mark أو تعريف شفرة لتستخدم تشفير غير UTF-8. ولا حظ حيث أن ASCII هي مجموعة إضافية من UTF-8 فإن كينونات ASCII العادية لا تحتاج تعريف تشفير بصورة صارمة.

إنه خطأ جسيم وفادح عندما تلتقي معالج XML بكينونة مع تشفير غير قادر على المعالجة

أمثلة تعريفات التشفير:

```
<?xml encoding='UTF-8'?>
<?xml encoding='EUC-JP'?>
```

٤-٤ معامل معالج XML للكينونات والمراجع

يلخص الجدول أسفل الكلام السياق التي قد تظهر في مراجع الأحرف والكينونة و invocations الخاصة بالكينونات غير الموزعة وسلوك معالج XML المتوقع والمطلوب في كل حالة. تصف العناوين في العمود الموجود جهة اليسار سياق التعريف:

المرجع في المحتوى كمرجع في أي مكان بعد علامة البدء وقبل علامة النهاية لعنصر وهي تماثل محتوى غير الطرفي.

المرجع في قيمة السمة كمرجع داخل إما قيمة سمة في علامة بدء أو قيمة افتراضية في تعريف سمة تماثل AttValue غير الطرفية.

حدوث كقيمة سمة مثل اسم Name وليس مرجع يظهر إما كقيمة لسمة تم إعلانها على أنها نوع ENTITY أو كواحدة من token المبعدة بمسافة في قيمة لسمة تم تعريفها على إنها نوع ENTITIES.

مرجع في قيمة كينونة كمرجع داخل معامل أو قيمة كينونة حرفية لقيمة داخلية في تعريف كينونة يماثل Entity Value غير طرفية.

مرجع في DTD كمرجع داخل إما المجموعة الإضافية الخارجية أو المجموعة الإضافية الداخلية التي تخص DTD ولكن خارج EntityValue أوAttValue.

نوع الكينونة	الحرف المعا	المعامل	داخلية عامة	خارجية موزعة عامة	غير موزعة
مرجع في المحتوى	غير معرف	مضمن	مضمن إذا كان صالح	ممنوع	مضمن
مرجع في قيمة سمة	غير معرف	مضمن حرفياً	ممنوع	ممنوع	مضمن
حدث كقيمة سمة	غير معرف	ممنوع	ممنوع	معرفا	غير معرف
مرجع قيمة الكينونة	مضمن في حرفياً	متجاوز عنة	متجاوز عنة	ممنوع	مضمن
مرجع في DTD	مضمن مثل PE	ممنوع	ممنوع	ممنوع	ممنوع

٤-٤-١ غير معرف

خارج DTD فإن حرف % ليس له أي ميزة خاصة ولهذا فما سيكون مراجع كينونة معامل في DTD لا يتم تعريفهم كترميز في المحتوى. بالمثل، أسماء الكينونات غير الموزعة لا يتم تعريفها إلا في حالة ظهورها في قيمة سمة معرفة بطريقة ملائمة.

٤-٤-٢ مضمن

يتم تضمين كينونة عندما يتم استرجاع ومعالجة النص الاستبدالي الخاص بها في مكان المرجع نفسه كما لو كانت جزء من المستند في الموقع الذي تم تعريف المرجع فيه. قد يحتوي النص الاستبدالى على كلاً من بيانات الحرف والترميز (ما عدا كينونات المعاملات) التي يجب تعريفها بالأسلوب المعتاد فيما عدا أن النص الاستبدالي للكينونات المستخدمة لتفادي محددات الترميز (كينونات amp, lt, gt, apos, quot) دائماً ما تم معاملتها كبيانات سلسلة AT&T تتسع إلى AT&T ولا يتم تعريف علامات مرجع حرف عندما يتم معالجة الحرف المشار إليه في مكان المرجع نفسه.

٤-٤-٣ تضمن إذا كانت صالحة

عندما يعرف معالج XML مرجع لكينونة موزعة من أجل التأكد من صلاحية مستند يجب أن نضمن المعالج النص الاستبدالي إذا كانت الكينونة خارجية وكان المعالج لا يحاول التأكد من صلاحية مستند XML يمكن للمعالج، لكنه ليس في حاجة إلى ذلك، تضمين نص استبدال الكينونة. إذا لم يضمن موزع غير صالح النص الاستبدالي يجب أن يعلم التطبيق أنه قد عرف ولكن لم يقرأ الكينونة.

تعتمد هذه القاعدة على التعريف أن التضمين الآلي المقدم بواسطة آلية كينونة SGML وXML والمصمم أساساً لدعم النمطية في النشر ليس بالضرورة ملائمة التطبيقات الأخرى وبالتحديد استعراض المستندات. فعلى سبيل المثال المستعرضات عندما تواجه مرجع كينونة موزع خارجياً قد تختار التقديم إشارة مرئية على وجود الكينونة واسترجاعها للعرض عند الطلب.

٤-٤-٤ ممنوع

ما يلي ممنوع ويكون خطأ جسيم:

- ظهور مرجع كينونة غير موزع
- ظهر أي حرف أو مرجع كينونة عام في DTD ما عدا داخل EntityValue أو AttValue.
- مرجع لكينونة خارجية في قيمة سمة.

٤-٤-٥ مضمنة في الأحرف

عندما يظهر مرجع كينونة في قيمة سمة أو تظهر مرجع كينونة معامل في قيمة كينونة حرفية يعالج نصها الاستبدالي في مكان المرجع نفسه كما لو كان جزء من المستند في الموقع الذي عرف فيه المرجع فيما عدا أن حرف علامة اقتباس زوجية أو فردية في النص الاستبدالي تعامل دائماً كحرف بيانات عادية فلا تنهي الحرفية. فعلى سبيل المثال ما يلي جيد التكوين:

```
<!ENTITY % YN '"Yes"' >
<!ENTITY What He Said "He said &YN;" >
```

أما هذا فليس جيد التكوين:

```
<!ENTITY EndAttr "27'" >
<element attribute='a-&EndAttr;>
```

٤-٤-٦ تبليغ

عندما يظهر اسم كينونة غير موزعة token في قيمة سمة لنوع معرف ENTITY أو ENTITIES يقوم معالج الصلاحية بإعلان تطبيق النظام والمعرفات العامة. فإن وجدت، في كلاً من الكينونة والتبليغ المصاحب لها.

٤-٤-٧ متجاوز عنه

عندما يظهر مرجع كينونة عام في EntityValue في تعريف كينونة يتم التجاوز عن وتركه كما هو.

٤-٤-٨ تضمين مثل PE

مثل الكينونات الموزعة الخارجية، تحتاج كينونات المعاملات إلى تضمينها فقط إذا كانت صالحة. عندما يتم تعريف مرجع كينونة معامل في DTD وتضمن. يتم تكبير نصها الاستبدالي بواسطة إرفاق حرف مسافة بادئة ومسافة لاحقة (#x20) الهدف هو تقيد النص الاستبدالي لكينونات المعامل لاحتواء عدد صحيح لصيغات برمجية tokens في DTD.

٤-٥ بناء نص استبدال الكائنات الداخلية

أثناء مناقشة معالجة الكينونات. الداخلية من المفيد التمييز بين نوعين من أنواع قيم الكينونات قيمة الكينونة الحرفية literal entity value هي السلسة بين علامات الاقتباس الموجود بالفعل في تعريف الكينونة وهي تماثل EntityValue غير الطرفية. النص الاستبدالي replacement text وهو محتوى الكينونة بعد استبدال مراجع الأحرف ومراجع كينونة المعامل.

قد تحتوي قيمة الكينونة الحرفية كما هي معطاة في تعريف كينونة داخلية (EntityValue) على أحرف، كينونة معامل، ومراجع كينونات عامة. مثل هذه المراجع يجب احتوائها كاملة داخل قيمة الكينونة الحرفية. النص الاستبدالي الفعلي المضمن كما هو موضح فيما سبق يجب أن يحتوي على replacement text النص الاستبدالي لأي كينونة معامل يشار إليها ويجب أن يحتوي الحرف المشار إليه، في مكان أي مراجع أحرف في قيمة الكينونة الحرفية ومع ذلك يجب ترك مراجع الكينونة العامة كما هي غير موسعة. المثال التالي يوضح ذلك.

```
<!ENTITY % pub    "&#xc9;ditions Gallimard" >
<!ENTITY   rights "All rights reserved" >
<!ENTITY   book   "La Peste: Albert Camus,
&#xA9; 1947 %pub;. &rights;" >
```

النص الاستبدالي لكينونة الكتاب book هي:

```
La Peste: Albert Camus,
(c) 1947 ƒditions Gallimard. &rights;
```

يتم توسيع مرجع الكينونة العامة &rights; عند ظهور المرجع &book; في محتوى المستند أو قيمة السمة.

قد تكون تلك القواعد البسيطة تفاعلات معقدة، للحصول على معلومات موسعة وتطبيقها على مثال صعب راجع "D توسيع مراجع الكائنات والأحرف".

٤-٦ الكينونات سابقة التعريف

يمكن استخدام كلا من مراجع الأحرف والكينونة لتفادي من أجل escape القوس جهة اليسار، علامة الضم ومحددات أخرى. يتم تحديد مجموعة من الكينونات العامة (amp, lt, gt, apos, quot) لهذا الفرص يمكن كذلك استخدام مراجع حرف عددية ويتم توسيعها مباشرة عند تعريفها

ويجب معالجتها كبيانات حرف حتى يمكن استخدام مراجع الأحرف العدديـــة ز&# ٦٠ و&#٣٨ لتفادي > و& عند ظهورهم في بيانات الأحرف.

يجب على كل معالجات XML تعريف تلك الكائنات سواء تم تعريفها أم لا لإمكانية التشـــغيل المتبادل، يجب أن تعلن تعرف مستندات XML الصالحة هذه الكينونـــات، مثـــل غيرهـــا، قبـــل استخدامها إذا تم تعريف تلك الكينونات يجب تعريفهم ككينونات داخلية يكون نصهم الاســـتبدالي حرف منفرد تم تفاديه أو مرجع حرف لهذا الحرف، كما يوضح ما يلي.

```
<!ENTITY lt    "#&38;#60;">
<!ENTITY gt    "#&62;">
<!ENTITY amp    "#&38;#38;">
<!ENTITY apos   "#&39;">
<!ENTITY quot   "#&34;">
```

لا حظ أن أحرف > و& في تعريف lt وamp يتم تفاديهم مرتين لتلبية متطلبـــات أن يكـــون استبدال النص جيد التكوين.

٤-٧ تعريفات مجموعة الرموز

تعرف مجموعة الرموز باسم تنسيق الكينونات غير الموزعة أو تنسيق العناصر إلى تحمل ســـمة مجموعة الرموز أو التطبيق الذي يوجه إلى تعليمات معالجه.

تقدم تعريفات مجموعة اسم لتلك المجموعة ليستخدم في تعريفات قائمة السمات والكينونة وفي مواصفات السمة كما تقدم معرف خارجي لمجموعة الرموز التي قد تسمح لمعالج XML أو تطبيق العميل الخاص به تحديد تطبيق مساعد قادر على معالجة البيانات في مجموعة رموز معطاة.

تعريفات مجموعة الرموز

```
[82] NotationDecl  ::= '<!NOTATION' S Name S (ExternalID
                        |  PublicID) S? '>'
[83] PublicID       ::= 'PUBLIC' S PubidLiteral
```

يجب أن تقدم معالجات XML تطبيقات باسم والمعرف أو المعرفات الخارجية لأي مجموعـــة رموز معرفة ومشار إليها في قيمة سمة أو تعريف سمة أو تعريف كينونة. وقد يحول المعـــرف الخارجي إلى معرف نظام أو اسم ملف أو أي معلومات أخرى مطلوبة للسماح للتطبيق باستدعاء

المعالج للبيانات في مجموعة الرموز الموضحة. (ليس من الخطأ المستندات XML أن تعرف وتشير إلى مجموعات رموز لا يتاح لها تطبيقات محددة مجموعة الرموز في النظام الذي يعمل علية معالج XML أو التطبيق).

٤-٨ كينونة مستند

يقوم كينونة المستند بلعب دور الجذر لشجرة الكينونة ونقطة البدء لمعالج XML هذه المواصفة لا تحدد كيف يتم وضع كينونة المستند بواسطة معالج XML على عكس الكينونات الأخرى، ليس لكينونة المستند اسم ويمكن أيضا أن تظهر على سلسلة إدخال معالج دون أي تعريف على الإطلاق.

٥. المطابقة

٥-١ معالجات الصلاحية وعدم الصلاحية

ينقسم معالجات مطابقة XML إلى فئتين هما: الصلاحية وعدم الصلاحية.

يجب أن يبلغ معالجات الصلاحية وعدم الصلاحية عن انتهاكات قيود مواصفات جودة التكوين في محتوى كينونة المستند وأي كينونات موزعة أخرى يقوموا بقراءتها.

يجب أن تبلغ معالجات الصلاحية عن انتهاكات القيود الموضحة بواسطة التعريفات في DTD وكذلك الفشل في تحقيق قيود الصلاحية المعطاة في المواصفات ولتحقيق ذلك يجب أن يقوم معالجات صلاحية XML بقراءة ومعالجة DTD بأكمله وكل الكينونات الخارجية الموزعة المشار إليها في المستند.

يطلب من معالجات عدم الصلاحية التحقيق من كينونة المستند فقط بما في ذلك مجموعة DTD الإضافية الداخلية الكاملة للتأكد من جودة التكوين وبينما لا يطلب منهم التحقيق من صلاحية المستند يطلب منهم معالجة كل التعريفات التي يقوموا بقراءتها في مجموعة DTD الإضافية الداخلية وفي أي كينونة معامل وصولاً إلى المرجع الأول لكينونة معامل لا يقوموا بقراءتها وهذا يعني أنك يجب على معالجها عدم الصلاحية استخدام المعلومات في تلك التعريفات لتسوية قيم السمة وتضمين النص الاستبدالي للكينونات الداخلية وكذلك تقديم قيم سمات افتراضية وفي نفس الوقت لا يجب ان يقوموا بمعالجة تعريفات كينونة أو قائمة سمة تظهر بعد مرجع لكينونة معامل لم تتم قراءته حيث قد تحتوي الكينونات على تعريفات تم التجاوز عنها.

٥-٢ استخدام معالجات XML

يمكن بسهولة توقع سلوك معالج صلاحية XML حيث يجب أن يقرأ كل جزء من المستند والإبلاغ عن كافة انتهاكات الصلاحية وجودة التكوين بينما يطلب من معالج عدم الصلاحية أقل من ذلـــك بكثير حيث لا تقرأ أي جزء من المستند بخلاف كينونة المستند ويكون لهذا تأثيرين لهم أهميتـــهم بالنسبة لمستخدمي معالجات XML.

- قد لا يستطيع معالج عدم الصلاحية اكتشاف بعض أخطاء جودة التكوين خاصـــة التـــي تتطلب قراءة كينونات خارجية وتتضمن الأمثلة القيود المسماة تعريف الكينونة والكينونة الموزعة No Recursion بالإضافة إلى بعض الحالات التي شرحها على أنها ممنوعـــة في "XML Processor Treatment of Entities and References٤.٤".
- تختلف المعلومات التي يتم تمريرها من المعالج إلى التطبيق بناء على ما إذا المعالج يقرأ الكينونات الخارجية والمعامل فعلى سبيل المثال، لا يقوم معالج عدم الصلاحية بتســوية قيم السمة افتراضية حيث يعتمد ذلك على إذا كان قد تم قراءة التعريفات فـــي كينونـــات المعامل أو الكينونات الخارجية.

للحصول على أقصي درجات إمكانية التشـــغيل المتداخـــل بيـــن معالجـــات XML المختلفـــة والتطبيقات التي تستخدم معالجات عدم الصلاحية لا يجب أن تعتمد على أي سلوك لا يطلب مـــن تلك المعالجات يجب أن تستخدم التطبيقات التي تتطلب استخدام السمات الافتراضية أو الكينونـــات الداخلية التي تعرف في الكينونات الخارجية معالجات صلاحية XML.

٦.مجموعة الرموز

تعطي الصيغة البرمجية الرسمية للغة XML في هـــذه المواصفـــة باســـتخدام مجمـــوع رمـــوز Extended Backus-Naur Form (EBNF) وتعرف كل قاعدة في الصيغة البرمجيـــة رمـــز واحد في النموذج.

Symbol ::= expression

تكتب الرموز بحرف استهلالي كبير في أول الكلمة في حالة تعريفها عن طريق تعبير عـــادي أو تكتب بحرف استهلالي صغير في أول الكلمة في أي حالة أخرى ويتم وضع السلاسل الحرفية بين علامات اقتباس.

يستخدم ما يلي في التعبير من الجهة اليمن للقاعدة لمطابقة السلاسل التي تتكون من حـــوف أو أكثر.

#xN

وحيث أن N هو عدد ست عشري صحيح يطابق هذا التعبير الحرف للموجود في ISO/IEC 10646 والذي تقدر قيمة التعليمات البرمجية (UCS-4) الخاصة به والمتفق عليها هي القيمة المشار إليها عند ترجمتها إلى عدد ثنائي لم يتم تحديده ويعتبر العدد للأصفار الأولية في نموذج #xN غير هام ويحكم عدد الأصفار الأولية في قيمة التعليمات البرمجية المماثلة عن طريق تشفير الأحرف المستخدمة وهي ليست ذات أهمية بارزة في XML.

[a-zA-Z], [#xN-#xN]

تطابق أي حرف له قيمة في النطاق أو النطاقات المحددة (شامل)

[^a-z], [^#xN-#xN]

تطابق أي حرف له قيمة خارج النطاق المحدد

[^abc], [^#xN#xN#xN]

تطابق أي حرف له قيمة ليست ضمن الأحرف المعطاة

"string"

تطابق سلسلة حرفية مطابقة لما تم إعطائه داخل علامات اقتباس مزدوجة

'string'

تطابق سلسلة حرفية مطابقة لما تم إعطائه داخل علامات اقتباس فردية

قد يتم دمج هذه الرموز لتطابق النماذج الأكثر تعقيداً كما يوضح ما يلي تمثل A وB تعبيرات بسيطة

(expression)

يعامل التعبير كوحدة ويمكن دمجه كما توضح القائمة التالية

A?

تطابق A أو لا تطابق شيء: A اختيارية

A B

تطابق A يتبعها B

A | B

تطابق A أوB ولكن ليس الاثنان معاً

A - B

تطابق أي سلسلة تطابق A ولا تطابق B

A+

تطابق حدوث أو أكثر للحرف A

A*

تطابق عدم حدوث حرف A أو حدوث الحرف A أكثر من مرة

بعض مجموعات الرموز الأخرى المستخدمة في الإنتاج

/* ... */

تعليق

[wfc: ...]

الرمز الإنجليزي السابق يرمز إلى قيد جودة التكوين: ويعرف هذا الرمز بالاسم قيد على جودة تكوين المستند يرتبط بإنتاج.

[vc: ...]

ترميز اللغة الإنجليزية السابقة إلى قيد الصلاحية وهو يعرف بالاسم قيد على المستندات الصالحة المرتبطة بإنتاج.

لاحق

أ المراجع

أ-١ المرجع الأساسية

IANA (Internet Assigned Numbers Authority). Official Names for Character Sets, ed. Keld Simonsen et al. See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets.

IETF RFC 1766 IETF (Internet Engineering Task Force). RFC 1766: Tags for the Identification of Languages, ed. H. Alvestrand. 1995.

ISO 639 (International Organization for Standardization). ISO 639:1988 (E). Code for the representation of names of languages. [Geneva]: International Organization for Standardization, 1988.

ISO 3166 (International Organization for Standardization). ISO 3166-1:1997 (E). Codes for the representation of names of countries and their subdivisions - Part 1: Country codes [Geneva]: International Organization for Standardization, 1997.

ISO/IEC 10646 ISO (International Organization for Standardization). ISO/IEC 10646-1993 (E). Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane. [Geneva]: International Organization for Standardization, 1993 (plus amendments AM 1 through AM 7).

Unicode The Unicode Consortium. The Unicode Standard, Version 2.0. Reading, Mass.: Addison-Wesley Developers Press, 1996.

أ-٢ مراجع أخرى

Aho/Ullman Aho, Alfred V., Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Reading: Addison-Wesley, 1986, rpt. corr. 1988.

Berners-Lee et al. Berners-Lee, T., R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax and Semantics. 1997. (Work in progress; see updates to RFC1738.)

Br ggemann-Klein Br ggemann-Klein, Anne. Regular Expressions into Finite Automata. Extended abstract in I. Simon, Hrsg., LATIN 1992, S. 97-98. Springer-Verlag, Berlin 1992. Full Version in Theoretical Computer Science 120: 197-213, 1993.

Br ggemann-Klein and Wood Br ggemann-Klein, Anne, and Derick Wood. Deterministic Regular Languages. Universit
f r Informatik, Bericht 38, Oktober 1991.

Clark James Clark. Comparison of SGML and XML. See http://www.w3.org/TR/ NOTE-sgml-xml-971215.

IETF RFC1738 IETF (Internet Engineering Task Force). RFC 1738: Uniform Resource Locators (URL), ed. T. Berners-Lee, L. Masinter, M. McCahill. 1994.

IETF RFC1808 IETF (Internet Engineering Task Force). RFC 1808: Relative Uniform Resource Locators, ed. R. Fielding. 1995.

IETF RFC2141 IETF (Internet Engineering Task Force). RFC 2141: URN Syntax, ed. R. Moats. 1997.

ISO 8879 ISO (International Organization for Standardization). ISO 8879:1986(E). Information processing - Text and Office Systems - Standard Generalized Markup Language (SGML). First edition –15-10-1986. [Geneva]: International Organization for Standardization, 1986.

ISO/IEC 10744 ISO (International Organization for Standardization). ISO/IEC 10744-1992 (E). Information technology - Hypermedia/Time-based Structuring Language (HyTime). [Geneva]: International Organization for Standardization, 1992. Extended Facilities Annexe. [Geneva]: International Organization for Standardization, 1996.

٠. فئات الأحرف

لي الصفات المعرفة في قياس Unicode يتم وضع الأحرف في فئات كأحرف أساسية من بينـــها لحروف الهجائية اللاتينية بدون علامات التشكيل وأحرف الرمزية أي الرموز المستخدمة للتعبير عن كلمة كاملة كما في اللغة الهيروغليفية والأحرف البادئة واللاحقة وتتضمن تلك الفئة من بيـــن أشياء أخرى معظم علامات التشكيل وتدمج تلك الفئات لتكوين فئة الأحرف كما يتم تميز الأرقـــام و الإضافات.

الأحرف

```
[84] Letter        ::= BaseChar | Ideographic
[85] BaseChar      ::= [#x0041-#x005A] | [#x0061-#x007A]
           | [#x00C0-#x00D6] | [#x00D8-#x00F6]
           | [#x00F8-#x00FF] | [#x0100-#x0131]
           | [#x0134-#x013E] | [#x0141-#x0148]
           | [#x014A-#x017E] | [#x0180-#x01C3]
           | [#x01CD-#x01F0] | [#x01F4-#x01F5]
           | [#x01FA-#x0217] | [#x0250-#x02A8]
           | [#x02BB-#x02C1] | #x0386 | [#x0388-#x038A]
           | #x038C | [#x038E-#x03A1] | [#x03A3-#x03CE]
           | [#x03D0-#x03D6] | #x03DA | #x03DC |
#x03DE
           | #x03E0 | [#x03E2-#x03F3] | [#x0401-#x040C]
           | [#x040E-#x044F] | [#x0451-#x045C]
           | [#x045E-#x0481] | [#x0490-#x04C4]
           | [#x04C7-#x04C8] | [#x04CB-#x04CC]
           | [#x04D0-#x04EB] | [#x04EE-#x04F5]
           | [#x04F8-#x04F9] | [#x0531-#x0556] | #x0559
           | [#x0561-#x0586] | [#x05D0-#x05EA]
```

| [#x05F0-#x05F2] | [#x0621-#x063A]
| [#x0641-#x064A] | [#x0671-#x06B7]
| [#x06BA-#x06BE] | [#x06C0-#x06CE]
| [#x06D0-#x06D3] | #x06D5 | [#x06E5-#x06E6]
| [#x0905-#x0939] | #x093D | [#x0958-#x0961]
| [#x0985-#x098C] | [#x098F-#x0990]
| [#x0993-#x09A8] | [#x09AA-#x09B0]
| #x09B2 | [#x09B6-#x09B9] | [#x09DC-#x09DD]
| [#x09DF-#x09E1] | [#x09F0-#x09F1]
| [#x0A05-#x0A0A] | [#x0A0F-#x0A10]
| [#x0A13-#x0A28] | [#x0A2A-#x0A30]
| [#x0A32-#x0A33] | [#x0A35-#x0A36]
| [#x0A38-#x0A39] | [#x0A59-#x0A5C]
| #x0A5E | [#x0A72-#x0A74] | [#x0A85-#x0A8B]
| #x0A8D | [#x0A8F-#x0A91] | [#x0A93-#x0AA8]
| [#x0AAA-#x0AB0] | [#x0AB2-#x0AB3]
| [#x0AB5-#x0AB9] | #x0ABD | #x0AE0
| [#x0B05-#x0B0C] | [#x0B0F-#x0B10]
| [#x0B13-#x0B28] | [#x0B2A-#x0B30]
| [#x0B32-#x0B33] | [#x0B36-#x0B39]
| #x0B3D | [#x0B5C-#x0B5D] | [#x0B5F-#x0B61]
| [#x0B85-#x0B8A] | [#x0B8E-#x0B90]
| [#x0B92-#x0B95] | [#x0B99-#x0B9A] | #x0B9C
| [#x0B9E-#x0B9F] | [#x0BA3-#x0BA4]
| [#x0BA8-#x0BAA] | [#x0BAE-#x0BB5]
| [#x0BB7-#x0BB9] | [#x0C05-#x0C0C]
| [#x0C0E-#x0C10] | [#x0C12-#x0C28]
| [#x0C2A-#x0C33] | [#x0C35-#x0C39]
| [#x0C60-#x0C61] | [#x0C85-#x0C8C]
| [#x0C8E-#x0C90] | [#x0C92-#x0CA8]

| [#x0CAA-#x0CB3] | [#x0CB5-#x0CB9] | #x0CDE
| [#x0CE0-#x0CE1] | [#x0D05-#x0D0C]
| [#x0D0E-#x0D10] | [#x0D12-#x0D28]
| [#x0D2A-#x0D39] | [#x0D60-#x0D61]
| [#x0E01-#x0E2E] | #x0E30 | [#x0E32-#x0E33]
| [#x0E40-#x0E45] | [#x0E81-#x0E82] | #x0E84
| [#x0E87-#x0E88] | #x0E8A | #x0E8D
| [#x0E94-#x0E97] | [#x0E99-#x0E9F]
| [#x0EA1-#x0EA3] | #x0EA5 | #x0EA7
| [#x0EAA-#x0EAB] | [#x0EAD-#x0EAE] | #x0EB0
| [#x0EB2-#x0EB3] | #x0EBD | [#x0EC0-#x0EC4]
| [#x0F40-#x0F47] | [#x0F49-#x0F69]
| [#x10A0-#x10C5] | [#x10D0-#x10F6] | #x1100
| [#x1102-#x1103] | [#x1105-#x1107] | #x1109
| [#x110B-#x110C] | [#x110E-#x1112] | #x113C
| #x113E | #x1140 | #x114C | #x114E | #x1150
| [#x1154-#x1155] | #x1159 | [#x115F-#x1161]
| #x1163 | #x1165 | #x1167 | #x1169
| [#x116D-#x116E] | [#x1172-#x1173] | #x1175
| #x119E | #x11A8 | #x11AB | [#x11AE-#x11AF]
| [#x11B7-#x11B8] | #x11BA | [#x11BC-#x11C2]
| #x11EB | #x11F0 | #x11F9 | [#x1E00-#x1E9B]
| [#x1EA0-#x1EF9] | [#x1F00-#x1F15]
| [#x1F18-#x1F1D] | [#x1F20-#x1F45]
| [#x1F48-#x1F4D] | [#x1F50-#x1F57] | #x1F59
| #x1F5B | #x1F5D | [#x1F5F-#x1F7D]
| [#x1F80-#x1FB4] | [#x1FB6-#x1FBC] | #x1FBE
| [#x1FC2-#x1FC4] | [#x1FC6-#x1FCC]
| [#x1FD0-#x1FD3] | [#x1FD6-#x1FDB]
| [#x1FE0-#x1FEC] | [#x1FF2-#x1FF4]
| [#x1FF6-#x1FFC] | #x2126 | [#x212A-#x212B]

| #x212E | [#x2180-#x2182] | [#x3041-#x3094]
| [#x30A1-#x30FA] | [#x3105-#x312C]
| [#xAC00-#xD7A3]

[86] Ideographic ::= [#x4E00-#x9FA5] | #x3007
| [#x3021-#x3029]

[87] CombiningChar ::= [#x0300-#x0345] | [#x0360-#x0361]
| [#x0483-#x0486] | [#x0591-#x05A1]
| [#x05A3-#x05B9] | [#x05BB-#x05BD]
| #x05BF | [#x05C1-#x05C2] | #x05C4
| [#x064B-#x0652] | #x0670
| [#x06D6-#x06DC] | [#x06DD-#x06DF]
| [#x06E0-#x06E4] | [#x06E7-#x06E8]
| [#x06EA-#x06ED] | [#x0901-#x0903]
| #x093C | [#x093E-#x094C] | #x094D
| [#x0951-#x0954] | [#x0962-#x0963]
| [#x0981-#x0983] | #x09BC | #x09BE
| #x09BF | [#x09C0-#x09C4]
| [#x09C7-#x09C8] | [#x09CB-#x09CD]
| #x09D7 | [#x09E2-#x09E3] | #x0A02
| #x0A3C | #x0A3E | #x0A3F
| [#x0A40-#x0A42] | [#x0A47-#x0A48]
| [#x0A4B-#x0A4D] | [#x0A70-#x0A71]
| [#x0A81-#x0A83] | #x0ABC



| [#x0ABE-#x0AC5] | [#x0AC7-#x0AC9]

| [#x0ACB-#x0ACD] | [#x0B01-#x0B03]

| #x0B3C | [#x0B3E-#x0B43]

| [#x0B47-#x0B48] | [#x0B4B-#x0B4D]

| [#x0B56-#x0B57] | [#x0B82-#x0B83]

| [#x0BBE-#x0BC2] | [#x0BC6-#x0BC8]

| [#x0BCA-#x0BCD] | #x0BD7

| [#x0C01-#x0C03] | [#x0C3E-#x0C44]

| [#x0C46-#x0C48] | [#x0C4A-#x0C4D]

| [#x0C55-#x0C56] | [#x0C82-#x0C83]

| [#x0CBE-#x0CC4] | [#x0CC6-#x0CC8]

| [#x0CCA-#x0CCD] | [#x0CD5-#x0CD6]

| [#x0D02-#x0D03] | [#x0D3E-#x0D43]

| [#x0D46-#x0D48] | [#x0D4A-#x0D4D]

| [#x0D57 | #x0E31 | [#x0E34-#x0E3A]

[88] Digit ::= [#x0030-#x0039] | [#x0660-#x0669]

| [#x06F0-#x06F9] | [#x0966-#x096F]

| [#x09E6-#x09EF] | [#x0A66-#x0A6F]

| [#x0AE6-#x0AEF] | [#x0B66-#x0B6F]

```
                                  | [#x0BE7-#x0BEF] | [#x0C66-
                                  #x0C6F]
                                  | [#x0CE6-#x0CEF] | [#x0D66-
                                  #x0D6F]
                                  | [#x0E50-#x0E59] | [#x0ED0-
                                  #x0ED9]
                                  | [#x0F20-#x0F29]
[89] Extender      ::=            #x00B7 | #x02D0 | #x02D1 |
#x0387
                                  | #x0640 | #x0E46 | #x0EC6 |
                                  #x3005
                                  | [#x3031-#x3035] | [#x309D-
                                  #x309E]
                                  | [#x30FC-#x30FE]
```

يمكن استنتاج فئات الأحرف المعرفة فيما سبق من قاعدة بيانات حرف Unicode كما يلي:

- يجب أن يكون لأحرف بداية الاسم إحدى فئات Ll, Lu, Lo, Lt, Nl.
- يجب أن يكون أحرف الاسم بخلاف أحرف بداية الاسم إحدى فئـــات Mc, Me, Mn, Lm, or Nd.
- لا يسمح بوجود الأحرف في مساحة التوافق التي تكون التعليمات البرمجية الخاصة بـــها أكبر من #xF900 وأصغر من #xFFFE في أسماء XML.
- لا يسمح بوجود الأحرف التي يكون لها حظ أو تكوين توافق أي الأحرف التي لها علامة تنسيق توافق في حقل ٥ من قاعدة البيانات ويرمز إليها بواسطة الحقل ٥ وتبدأ بعلامـــة <
- تعالج الأحرف التالية كأحرف بداية الاسم بدلا من أحرف الاســم لأن ملــف الخاصيــة يصنفهما مثل [#x02BB-#x02C1], #x0559, #x06E5, #x06E6.
- يتم استبعاد أحرف #x20DD-#x20E0 بما يتماشى مع Unicode المقطع 5.14
- يصنف حرف #x00B7 كإضافات لأن قائمة الخاصية تعريفها كذلك.
- يضاف حرف #x0387 كاسم حرف لأن #x00B7 هي الحرف المتفــق عليــه الــذي يساويها.
- يسمح بوجود الأحرف ' : ' و' _ ' كأحرف بداية الاسم.

♦ يسمح بوجود الأحرف '.' و' _ ' كأحرف الاسم.

ج لغة XML وSGML (غير أساسية)

تصمم لغة XML لتكون مجموعة إضافية من SGML بمعني أن يكون كل مستند XML صالح متوافق مع مستند SGML وللحصول على مقارنة مفصلة عن القيود الإضافية التي تضعها لغة XML على المستندات غير مستندات SGML راجع [Clark].

د مراجع الكينونة والحرف (غير أساسيي)

يحتوى هذا الملحق على بعض الأمثلة التي توضح ترتيب توسع وتعريف مرجع الحرف والكينونة كما هو محدد في "٤.٤"XML Processor Treatment of Entities and References".

إذا احتواء DTD على تعريف

```
<!ENTITY example "<p>An ampersand (#&38;#38;) may be escaped
numerically (#&38;#38;#38;) or with a general entity
(&amp;amp;).</p>" >
```

سيقوم معالج XML بتعريف مراجع الحرف عند توزيعه لتعريف الكيان ويحلهم قبل تخزين السلسلة التالية كقيمة لكينونة "example":

```
<p>An ampersand (#&38;) may be escaped
numerically (#&38;#38;) or with a general entity
(&amp;amp;).</p>
```

يسبب وجود مرجع في المستند يشير إلى &example إلى إعادة توزيع النص وفي نفس الوقت علامات بداية ونهاية عنصر أ ويتم تعريف الثلاث مراجع كما يتم توسيعهما ويكون الناتج عنصر أ بالمحتويات التالية (كل البيانات دون محددات أو ترميز):

```
An ampersand (&) may be escaped
numerically (#&38;) or with a general entity
(&amp;).
```

يعرض المثال التالي القواعد وتأثيرها لاحظ أن أرقام السطور في هذا المثال تستخدم كمرجع فقط.

```
1 <?xml version='1.0'?>
2 <!DOCTYPE test [
3 <!ELEMENT test (#PCDATA) >
4 <!ENTITY % xx '#&37;zz;'>
5 <!ENTITY % zz '#&60;!ENTITY tricky "error-prone" >' >
6 %xx;
7 ]>
8 <test>This sample shows a &tricky; method.</test>
```

ينتج ما يلي كما سبق

في السطر الرابع: يتم توسيع المرجع إلى الحرف ٣٧ فوراً ويتم تخزين كينونة معامل xx في جدول الرمز بقيمة zz% وحيث أن النص الاستبدالي لا يعاد مسحة ضوئية لا يتم تعريف المرجع إلى كينونة معامل zz ويكون ذلك خطأ حيث أن zz لم يعلن عنة بعد.

في السطر الخامس: يتم توسيع مرجع حرف &#٦٠ فوراً وتخزن كينونة معامل zz مع النص الاستبدالي <!ENTITY tricky error-prone > وهو تعريف كينونة جيد التكوين.

في السطر السادس: يتم تعريف المرجع إلى xx ويوزع النص الاستبدالي لكينونة معامل xx (هي zz%) ويتم تعريف المرجع إلى zz في وقته ويوزع النص الاستبدالي له وهو (<!ENTITY tricky error-prone >) وبذلك يكون قد تم تعريف الكينونة العامة tricky مع النص الاستبدالي error-prone.

في السطر الثامن: يتم تعريف المرجع إلى الكينونة العامة tricky ويتم توسيعها وبذلك يكون المحتوى الكامل لعنصر test هو السلسلة التي تشرح نفسها ولا تتبع صيغة نحوية This sample shows a error-prone method.

هـ- محددات أنواع المحتوى (غير أساسي)

للحصول على التوافق من المطلوب أن تكون أنواع المحتوى في تعريف نوع العنصر محددة.

تتطلب SGML أنواع محتوى محددة وتطلق عليهم SGML غير واضحة unambiguous وقد تشير معالجات XML المبينة باستخدام أنظمة SGML إلى أنواع المحتوى غير المحددة على أنها أخطاء.

فعلى سبيل المثال يعتبر نوع المحتوي ((b, c) | (b, d)) غير محدد والسبب في ذلك وجود حرف b في البداية لا يجعل الموزع يعرف أي حرف b في النموذج يتم مطابقته دون النظر للعنصر الذي ينتج b في تلك الحالة يمكن طي المرجعين إلى b في مرجع واحد وبذلك يقرأ النموذج أو ((b, (c | d) وبذلك تطابق b بادئة فقط اسم واحد في نموذج المحتوى ولا يجب على الموزع البحث عن ما يلي العنصر b فسواء كانت c أو d فهي مقبولة.

بصيغة أكثر رسمية: يمكن بناء QUT ONTENT حالة نهائية من نموذج المحتوى باستخدام العمليات الحسابية القياسية مثل: عمليات حسابية [Aho/Ullman] Ullman. في التعبير العادي (بمعنى كل ورقة عقده في syntax الشجرة لتعبير العادي) إذا كان لأي موقع مجموعة تابعة يسمى فيها أكثر من موقع تابع بنفس اسم نوع العنصر يكون نموذج المحتوى فيه خطأ ويتم الإبلاغ عند كخطأ.

ينتج وجود العمليات الحسابية العديد ولكن ليس كل نماذج المحتويات غير المحددة أن يتم تقليلها بصورة آلية إلى نماذج محددة مساوية راجع Br ggemann Klein 1991 .[Br ggemann-Klein]

و.التتبع الآلي لترميزات الأحرف (غير معياري)

تعمل تعريفات تشفير XML كتسمية داخلية على كل كينونة مشيرة إلى أي تشفير الأحرف يكون مستخدم. قبل أن يستطيع معالج XML قراءة تسمية داخلية يجب عليه أي تشفيرات الأحرف يكون مستخدما وهو ما تحاول التسمية الداخلية الإشارة إليه في الحالة العامة فيكون هذا وضعاً غاية في الصعوبة ولكن ليس ميئوساً من في XML لأن XML تحدد الحالة العامة بطريقتين هما: يفترض في كل تطبيق تدعم تشفيرات مجموعة أحرف نهائية واحدة فقط ويفيد تعريف ترميز XML في الموقع والمحتوى من أجل جعل position لتتبع تشفير الحرف المُستخدم في كل كينونة في الحالات العادية آليا أيضا في العديد من الحالات تتاح مصادر معلومات أخرى بالإضافة إلى إرسال بيانات XML نفسها يمكن تميز حالتين بناء على ما إذا كانت كينونة XML مقدمة للمعالج بدون أو لمعلومات (خارجية) مصاحبة راجع الحالة الأولى أولا.

لأن كل كينونة XML لا تكون في تنسيق UTF-8 أو UTF-16 يجب أن تبدأ بتعريف شفرة XML حيث يجب أن تكون الأحرف الأولى <?xml يمكن لأي معالج تأكيد تحديد بعد اثنين أو أربعة مثمنات من المدخلات، أي الحالات التالية يتم تطبيقها وأثناء قراءة القائمة قد يساعد معرفة

أن UCS-4 هو #x0000003C و ? هي #x0000003F وتكون Byte Order Mark المطلوبة من إرسال بيانات UTF-16 هي #xFEFF.

- 00 00 00 3C: UCS-4, big-endian machine (1234 order)
- 3C 00 00 00: UCS-4, little-endian machine (4321 order)
- 00 00 3C 00: UCS-4, unusual octet order (2143)
- 00 3C 00 00: UCS-4, unusual octet order (3412)
- FE FF: UTF-16, big-endian
- FF FE: UTF-16, little-endian
- 00 3C 00 3F: UTF-16, big-endian, no Byte Order Mark (and thus, strictly
- 3C 00 3F 00: UTF-16, little-endian, no Byte Order Mark (and thus, strictly
- 3C 3F 78 6D: UTF-8, ISO 646, ASCII, some part of ISO بعض أجزاء من 8859 أو أي ٧ بت أو ٨ بت أو ترميز مختلط العرض مما يؤكد أن أحرف ASCII لها موضعها العادي، العرض والقيم يجب قراءة تعريف التشفير الفعلي لتحديد أي من هـذا يطبق ولكن حيث أن كل هذه الشفرات تستخدم نفس نموذج بت لأحرف ASCII يمكـن قراءة تعريف الشفرات بطريقة صحيحة.
- 4C 6F A7 94: EBCDIC (من بعض النواحي يجب قراءة تعريـف التشـفير كـاملاً لنستطيع تحديد أي صفحات التعليمات البرمجية يتم استخدامها).
- أخرى: UTF-8 بدون تعريف تشفير أو يكون إرسال البيانات فاسد جزئي أو مضمن فـي wrapper من نوع ما.

يكفي هذا المستوى من التتبع الآلي لقراءة تعريف تشفير XML وتوزيع معرف تشفير الحـوف التي مازالت ضرورية لتمييز الأعضاء الفرديين لكل عائلة من الشفرات (مثـل لتعلـم UTF-8 from 8859 وأجزاء 8859 من بعضها البعض أو لتميز صفحة تعليمـات EBCDIC الـبرمجة المحددة المستخدمة)

لأن محتويات تعريف التشفير مقيدة لأحرف ASCII يمكن لمعالج قراءة تعريف الترميز كاملاً بمجرد تحديد أي عائد مجموعة من الشفرات تستخدم في التطبيق العلمي تقع شفرات الأحرف الشائعة الاستخدام في أحد الفئات السابقة يتيح تعريف تشفير XML تسمية معقولة يعتمد عليها في الحدود المتاحة لشفرات الأحرف حتى ولو كانت مصادر خارجية للمعلومات عند نظام التشغيل أو مستوى بروتوكول النقل لا يعتمد عليها.

بمجرد أن يحدد المعالج تشفير الحرف المستخدم يمكن أن يعمل بصورة مناسبة سواء عن طريق وضع توجيه روتين إدخال منفصل لكل حالة أو عن طريق استدعاء وظيفة التحويل المناسبة لكل حرف إدخال.

مثل أي نظام تسمية آلي لن يعمل تعريف شفرات XML إذا قام أي برنامج بتغير مجموعة أو ترميز أحرف الكينونة دون تحديث تعريف الشفرات يجب أن يكون منفذي روتين تشفير الأحرف حذرين في تأكد دقة المعلومات الخارجية والداخلية المستخدمة لتسمية الكينونة.

تحدث الحالة المتاحة الثانية عندما تصاحب كينونة XML بمعلومات تشفير كما بعض أنظمة الملفات وبعض بروتوكولات شبكات العمل عندما تتاح عدة مصادر للمعلومات تكون الأولية النسبية والأسلوب الأمثل لمعالجة التعارضات محددة كجزء من البروتوكول المستوى الأعلى المستخدم في تسليم XML يجب أن تكون قواعد الأولية النسبية للتسمية الداخلية وتسمية نوع MIME في تسمية رأسية خارجية على سبيل المثال جزء من مستند RFC الذي يعرف أنواع MIME النصي text/xml /xml وتطبيق application/xml لإمكانية التشغيل المتبادل يفضل استخدام القواعد التالية.

- إذا كانت كينونة XML في ملف تستخدم Byte-Order Mark وتعريف تشفير PI (إن وجد) لتحديد تشفير الحرف كل المصادر وheuristics الأخرى للمعلومات هي لمعالجة الخطأ فقط.
- إذا تم تسليم كينونة XML مع نوع MIME لنص text/xml xml فإن معامل CHARSET على نوع MIME يحدد طريقة تشفير الأحرف وتكون كل heuristics ومصادر المعلومات لمعالجة الأخطاء فقط.
- إذا تم تسليم كينونة XML مع نوع MIME تطبيق (application/xml) تستخدم Byte-Order Mark وتعريف تشفير PI (إن وجد) لتحديد تشغيل الحرف. كل مصادر heuristics المعلومات الأخرى تكون لتصحيح الأخطاء فقط.

تطبيق تلك القواعد فقط في غياب الوثيق على مستوى البروتوكول وبالتحديد عند تعريف أنواع MIME نص text/xml وتطبيق application/xml وبذلك تحل التوصيات التي تخص RFC المرتبطة محل هذه القواعد.

مجموعة عمل G. W3C XML "غير معياري"

تم الإعداد والموافقة على نشر هذه المواصفة بواسطة مجموعة عمل W3C XML لا تعني موافقة مجموعة عمل هذه المواصفة أن كل أعضاء مجموعة العمل قد أعضاء مجموعة العمل قد أعطوا أصواتهم لصالح هذه الموافقة. أعضاء مجموعة عمل XML الحاليين والسابقين هم.

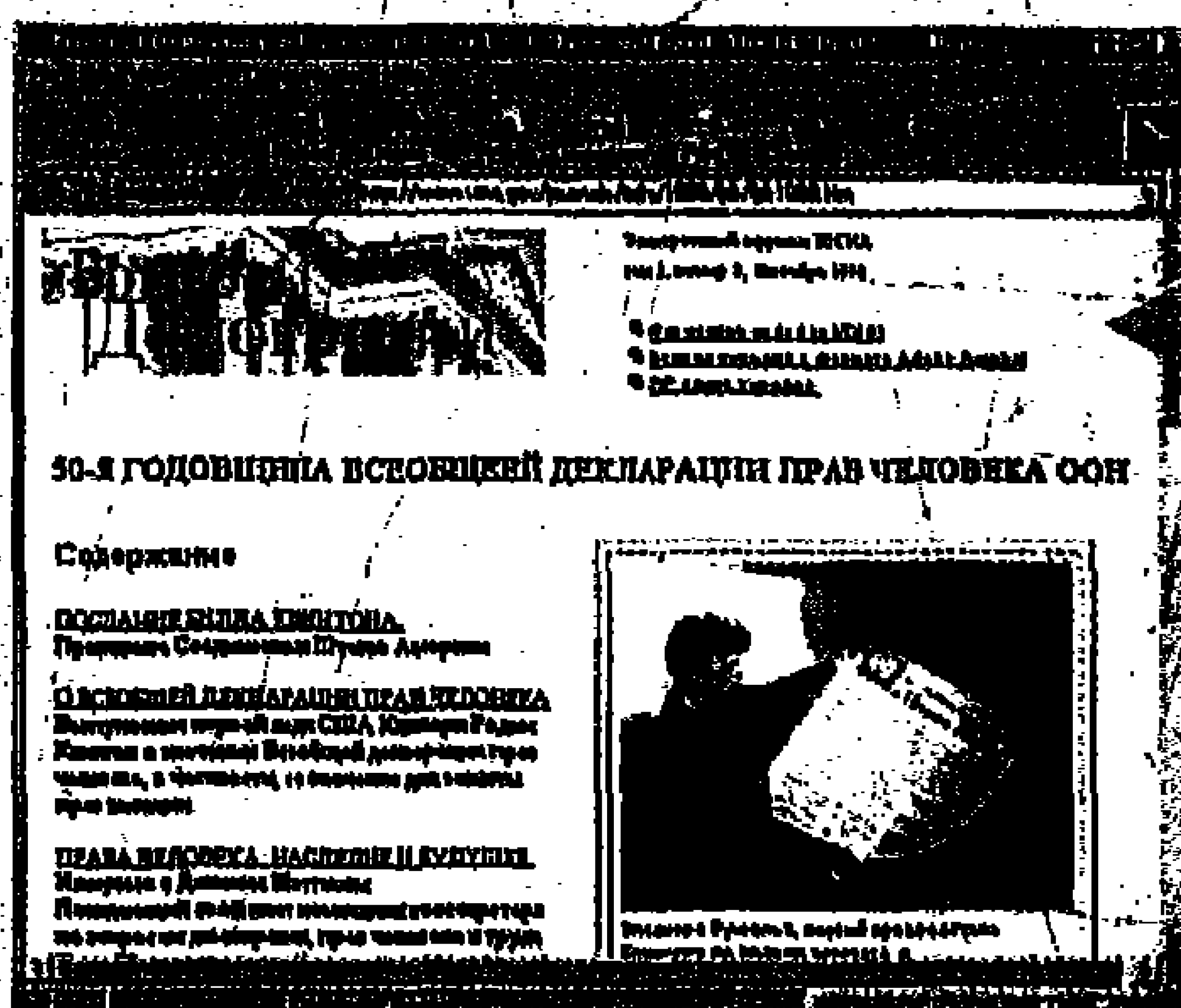
Jon Bosak, Sun (Chair); James Clark (Technical Lead); Tim Bray, Textuality and Netscape (XML Co-editor); Jean Paoli, Microsoft (XML Co-editor); C. M. Sperberg-McQueen, U. of Ill. (XML Co-editor); Dan Connolly, W3C (W3C Liaison); Paula Angerstein, Texcel; Steve DeRose, INSO; Dave Hollander, HP; Eliot Kimber, ISOGEN; Eve Maler, ArborText; Tom Magliery, NCSA; Murray Maloney, Muzmo and Grif; Makoto Murata, Fuji Xerox Information Systems; Joel Nava, Adobe; Conleth O"Connell, Vignette; Peter Sharpe, SoftQuad; John Tigue, DataChannel

If XML can do it, you can do it too ...

XML is revolutionizing Web-site development by making difficult tasks easier - and many new tasks possible. Based on the latest W3C standards, this thorough tutorial-plus-reference takes you step by step through everything you need to know to put XML to work, form the fundamentals of the XML Language to document construction and simple XML-based solutions for specialized markup problems. Numerous examples, specifications, and addresses for relevant Web sites leave no questions unanswered.

Inside, you'll find complete coverage of XML

- Follow simple rules to create well-formed XML documents
- Define tags that make sense for your document
- Format your documents with style sheets
- Create your own markup languages
- Validate documents with DTDs
- Explore RDF,VML, CDF, and other XML applications.



Write Web pages in foreign languages and diverse scripts

browse XML documents on the Web

Atlanta Braves

Batters

Player	P	G	GS	AB	R	H	D	T	HR	RBI	S	CS	SH	SF	E	BB	SO	HBP
Marty Malloy	Second Base	11	8	28	3	5	1	0	1	1	0	0	0	0	0	2	2	0
Ozzie Guillen	Shortstop	83	59	264	35	73	15	1	1	22	1	4	4	2	6	24	25	1
Danny Bautista	Outfield	82	27	144	17	36	11	0	3	17	1	0	3	2	2	7	21	0
Gerald Williams	Outfield	129	51	266	46	81	18	3	10	44	11	5	2	1	5	17	48	3
Javier Lopez	Catcher	133	124	489	73	139	21	1	34	106	5	3	1	8	5	30	85	6
Ryan Klesko	Outfield	129	124	427	69	117	29	1	18	70	5	3	0	4	2	56	66	2
Andres Galarraga	First Base	153	151	555	103	169	27	1	44	121	7	6	0	5	11	63	146	25
Wes Helms	Third Base	7	2	13	2	4	1	0	1	2	0	0	0	0	1	0	4	0

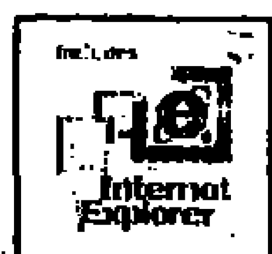
Pitchers

Player	P	G	GS	W	L	S	CG	SO	ERA	IP	HR	R	ER	HB	WP	B	BB	K
Tom Glavine	Starting Pitcher	33	33	20	6	0	4	3	2.47	229.1	13	67	63	2		0	74	157

Florida Marlins



Essential XML tools and code samples on CD-ROM include :



- code for all numbered listing in the book
- XML browsers and tools
- World Wide Web Consortium XML standards



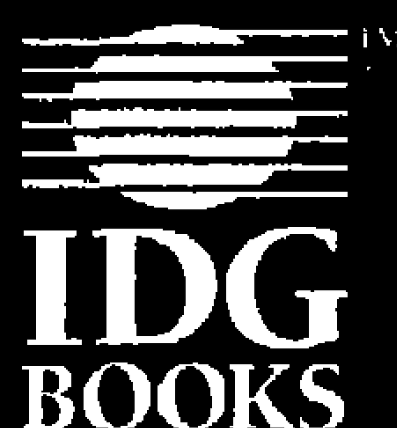
Shareware programs are fully functional, free trial versions of copyrighted programs. If you like particular programs, register with their authors for anominal fee and receive licenses, enhanced versions, and technical support. Freeware programs are free, copyrighted games, applications, and utilities. you can copy them to as many PCs as you like - free- but they have no technical support.

System Requirements: Java 1.1 or later compatible platform

Reader Level: Beginning to Advanced

Shelving Category: Internet/Programming/XML





الوكيل الوحيد على مستوى الشرق الأوسط
دار الفاروق للنشر والتوزيع
DAR EL FAROUK
فرع وسط البلد: ٣٠ شارع منصور المبتديان متفرع من شارع مجلس الشعب
بجانب محطة مترو سعد زغلول - القاهرة - مصر
تليفون: ٠٠٢/٠٢/٧٩٥٢٠٢٢ - ٠٠٢/٠٢/٧٩٤٢٢٠٢ - فاكس: ٠٠٢/٠٢/٧٩٤٣٦٤٢
فرع الدقي (الإدارة): ١١ شارع الدقي - الدور السابع - منزل كوبري الدقي - إتجاه الجامعة

ISBN 0-7645-3236-7

